

情報技術者試験受験テキスト

基礎理論

アルゴリズム



令和元年 10 月発行

KMC 学習所

txt0102 アルゴリズム目次

txt01021	データ構造	-04-
txt010211	配列	-04-
①	一次元配列	-04-
②	二次元配列・三次元配列	-04-
③	配列を利用した線形探索	-05-
④	配列を利用した直接探索・二分探索	-07-
⑤	配列のデータ更新と削除	-07-
⑥	配列のデータ挿入	-08-
txt010212	リスト	-13-
①	リストの定義と特徴	-13-
②	リストを利用したデータ探索	-13-
③	リストのデータ更新と削除	-14-
④	リストのデータ挿入	-15-
txt010213	スタックと待ち行列	-22-
①	スタックの定義と特徴	-22-
②	スタックの操作	-22-
③	スタックの応用	-23-
④	待ち行列の定義と特徴	-25-
⑤	待ち行列の操作	-25-
txt010214	二分木と二分探索木	-32-
①	二分木の定義と表現	-32-
②	二分木の構成	-32-
③	二分探索木の定義	-33-
④	二分探索木の探索	-33-
⑤	二分探索木の挿入	-34-
⑥	二分探索木の削除	-35-
txt010215	二分木の巡回法とその応用	-39-
①	二分木の巡回法	-39-
②	二分木と四則演算	-40-
③	逆ポーランド記法への活用	-40-

txt010216	バランズ木とヒープ木	-49-
①	バランズ木と完全二分木	-49-
②	多分木	-50-
③	5 次のバランズ木	-50-
④	ヒープ木の定義と操作	-51-
txt01022	アルゴリズム	-57-
txt010221	探索アルゴリズム	-57-
①	線形探索アルゴリズム	-57-
②	二分探索あるゴリズム	-59-
③	ハッシュ法	-61-
④	アルゴリズムと実行効率	-63-
⑤	探索法と最大比較回数	-63-
txt010222	整列アルゴリズム	-71-
①	整列のアルゴリズム	-71-
②	基本交換法	-71-
③	基本挿入法	-72-
④	基本選択法	-73-
⑤	基本整列アルゴリズムの流れ図	-75-
txt010223	高速整列法	-83-
①	ヒープソート	-83-
②	クイックソート	-83-
③	シェルソート	-86-
④	シェーカーソート	-87-
⑤	バブルソートの計算量	-88-
⑥	クイックソートの計算量	-89-
txt010224	単一ファイル処理	-94-
①	単一ファイル処理	-94-
②	単一ファイル処理の基本パターン	-96-
③	グループ集計とコントロールブレーク処理	-97-
txt010225	複数ファイル処理	-100-
①	複数ファイル処理	-100-
②	複数ファイル処理の基本	-101-

③ ファイルの併合	-101-
④ ファイルの突合せ	-103-
⑤ ファイルの更新	-104-
⑥ マスタファイルへのレコードの追加・削除	-105-
 txt010226 文字列の探索・圧縮・置換	-108-
① 力任せの探索法	-108-
② ボイヤ・ムーアの方法	-110-
③ 文字列の圧縮	-113-
④ 文字列の置換	-115-
 txt010227 再帰・バックス記法・正規表現	-127-
① 再帰アルゴリズム	-127-
② バックス記法	-128-
③ 正規表現・分割統治法	-129-

txt0102 アルゴリズム

txt01021 データ構造

txt010211 配列

① 一次元配列

① 配列の定義と表現

ア 定義

配列は同じデータ型、同じ大きさのデータが物理的に連続して並んでいるデータ構造をもっている。配列名を有し、個々のデータを表す配列の要素で構成され、定義された一定の大きさNをもつ。

イ 表現

配列の特定の要素は配列名の後ろに()とその中に添字を記入して表す。配列名をT B L、先頭の添字が1の場合、I番目の添字の要素はT B L(I)で表す。

② 配列の要素

配列の要素に含まれる各データを添字によって特定できる。添字は配列の先頭から何番目の要素であるかを表している。添字の先頭は0または1から始まる。大きさ7の配列T B L(I)の各要素にA、B、C、…、Gの各文字を格納した例を示す。

添字	1	2	3	4	5	6	7
配列 T B L (I)	A	B	C	D	E	F	G

配列の大きさ 7

② 二次元配列・三次元配列

① 二次元配列

二次元配列は2つの軸を考えて、それぞれの軸上に特定の数値を設定し、その数値の交点に対応する値を設定する考え方である。同様に考えて、3軸を用いて、3軸の値に対応する交点の数値を設定すると三次元配列となり、4以上の多軸を考え、それぞれの軸の値の交点に対応する値を設定すると多次元配列となる。

二次元配列は、行の要素番号を I 、列の要素番号を J で表すと、 I 、 J で決まる配列の要素は $TBL(I, J)$ で表すことができる。二次元配列 $TBL(I, J)$ を利用して、行方向の合計、列方向の合計、全体の合計を求めることができる。

(1. 1)	(1. 2)	...	(1. j)	...	(1. n)
(2. 1)	(2. 2)				
(i. 1)			(i. j)		(i. n)
(m. 1)			(m. j)		(m. n)

⑥ 一次元配列への変換

M 行 N 列の二次元配列 $TBL(I, J)$ を一次元配列 $TBL(K)$ に変換する場合、次の式を利用し、一次元配列の添字 K を求める。

$$K = (I - 1) \times M + J$$

⑦ 三次元の配列の要素と変換

三次元配列は $TBL(I, J, K)$ で表す。この場合の I は面、 J は行、 K は列を表す。

L 面 M 行 N 列の三次元配列 $TBL(I, J, K)$ を一次元配列 $TBL(P)$ に変換する場合、次の式を利用し、一次元配列の添字 P を求める。

$$P = (I - 1) \times M \times N + (J - 1) \times M + K$$

③ 配列を利用した線形探索

① 探索方法

配列を利用した探索の方法には、次の 3 通りの方法がある。

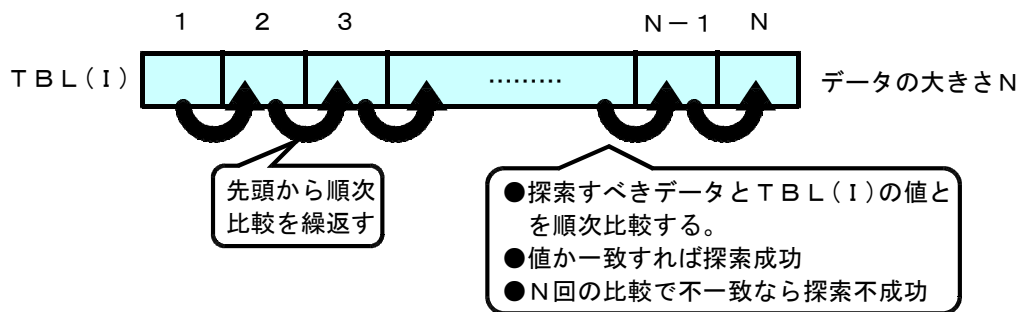
㊦ 線形探索

線形探索には、通常の線形探索と番兵を使用した線形探索がある。

㊩ 二分探索

㊫ 直接探索

⑥ 通常の線形探索



通常の線形探索は、配列の先頭または後方から配列の要素と探索すべきデータを逐次比較して、要素の内容と探索データが等しくなる添字を求める方法である。

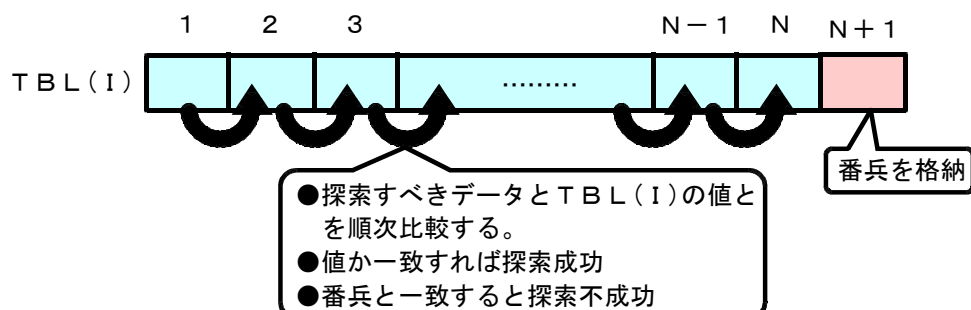
⑦ 通常の線形探索の手順

通常の線形探索の手順は次の通りである。

- ㊦ 添字 I を 1 (または N) で初期化する。
- ㊧ 添字 I が配列の大きさ N を超える (または 1 より小さくなる) と、探索データが配列内に存在しないため探索を終了する。添字 I が配列の大きさ N 以下 (または 1 以上) ならば㊨～㊩の処理を繰り返す。
- ㊨ 配列の要素 $TBL(I)$ と探索データ K を比較する。
- ㊩ $TBL(I) = K$ ならば探索成功。処理を終了する。
- ㊪ $TBL(I) \neq K$ ならば探索不成功。 $I + 1 \rightarrow I$ (または $I - 1 \rightarrow I$) を実行し、㊧に戻る。

⑧ 番兵を利用した線形探索

番兵を利用した探索は大きさ N の配列の $N + 1$ 番目の要素 (図の赤の要素) に探索するデータを格納する方法である。番兵を利用すると、比較回数を減少させることができる。



⑨ 線形探索の終了条件

配列を利用して線形探索のような処理を行う場合、処理の終了条件が必要になる。線形探索を行う場合、次のような終了条件が考えられる。

- ㊦ 配列の前方から探索する場合、定義された配列の大きさNを超えると終了する。
- ㊧ 配列の後方から探索する場合、先頭の要素番号より小さくなると終了する。
- ㊨ 配列の大きさが未知の場合、番兵を設定し、番兵に達すると終了する。

④ 配列を利用した直接探索・二分探索

㊱ 直接探索

探索すべき位置が分かっている場合には、その位置を示す添字の値を用いて直接探索する。二つの配列を使用して、一方の配列で線形探索を用いて添字 I を求め、その添字 I を用いてもう一方の配列を直接探索し目的のデータを得ることができる。

㊲ 二分探索

データが昇順または降順に並んでいる場合には、二分探索法を用いて目的のデータを探索することができる。昇順に並んでいる場合、二分探索は中央の要素の値と探索データを比較し、等しければ探索成功、探索データが小さい場合、中央の要素より小さい範囲を新しい範囲とし、探索データが大きい場合、中央の要素より大きい範囲を新しい範囲として探索を繰り返す方法である。探索回数は線形探索と比較して、平均探索回数、最大探索回数共に小さくなる。

⑤ 配列のデータ更新と削除

㊳ データの更新

データの更新は、特定のデータを探索後、そのデータの内容を読み出し、内容を変更し、書き込む一連の操作である。探索には線形探索、直接探索、二分探索のいずれかを使用する。

㊴ データの削除

データの削除は、削除すべき配列の要素を探索法を利用して求め削除する。削除の方法には物理的削除と論理的削除がある。

物理的削除は、メモリ上からデータの内容を除去することであり、論理的な削除は、削除する要素の内容を削除されている意味の内容で書き換えることである。論理的に削除されたデータは使用できないが、メモリ上には残っている。

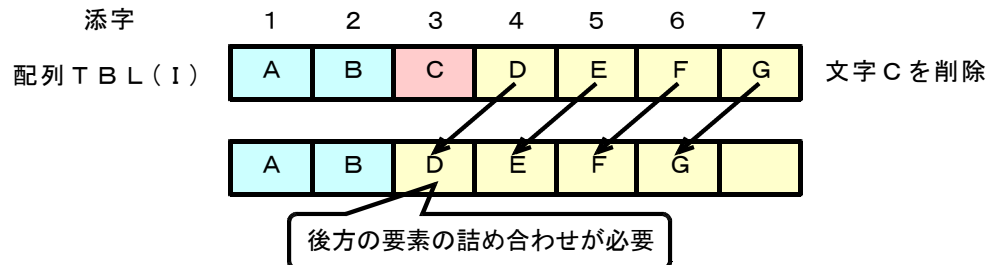
㊵ 物理的削除の手順

物理的削除は、削除した要素の空白部分を無くするため、順次配列の要素を前に詰めていく処理が必要である。

大きさNの配列 T B L (I) の添字 K から後方の各要素を 1 要素前に移動させる手順は次のようになる。

- ㊦ IにKを格納する。
- ㊧ $TBL(I) \rightarrow TBL(I-1)$ を実行する。
- ㊨ $I+1 \rightarrow I$ を実行する。
- ㊩ $I > N$ になると終了する。 $I \leq N$ ならば㊧に戻る。

M文字前に移動する操作は㊧の操作が $TBL(I) \rightarrow TBL(I-M)$ となる。

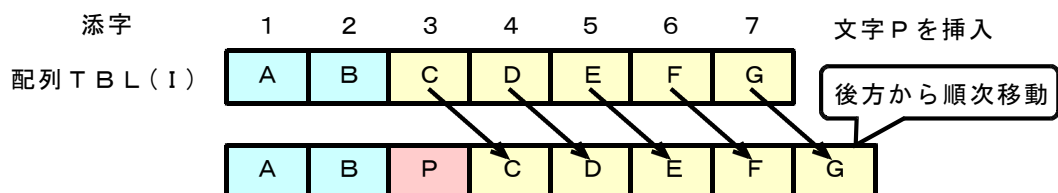


⑥ 配列のデータ挿入

㊐ データ挿入

データの挿入は、挿入位置を探索法を利用して求め、挿入に必要な空白部を設ける。挿入位置の探索には、線形探索、直接探索、二分探索を利用する。

空白部を設ける場合、配列の要素の移動は配列の後方から挿入位置Kまでの要素について実行する。



㊑ データ挿入手順

大きさNの配列 $TBL(I)$ の添字Kの位置に1文字を挿入する手順

- ㊦ IにNを格納する。
- ㊧ $TBL(I) \rightarrow TBL(I+1)$ を実行する。
- ㊨ $I-1 \rightarrow I$ を実行する。
- ㊩ $I \geq K$ ならば㊧に戻る。
- ㊪ $I < K$ になると、 $I=K$ の位置に新しい文字を格納する。
- ㊫ 配列の大きさを $N+1 \rightarrow N$ で修正する。

M個のデータを挿入する場合、1個の場合の操作手順の次の箇所を修正する。

①の操作が $TBL(I) \rightarrow TBL(I+M)$ となる。

②のM個のデータを格納する操作は、 $TBL(K)$ から $TBL(K+M-1)$ にM個の文字を格納する。

③のM個のデータを挿入した配列の大きさは $N+M \rightarrow N$ となる。

例題演習

次のうち配列の特徴を表している適切な組合せはどれか。

- I 一つの要素の大きさはあらかじめ決まっている。
- II 配列全体の大きさは実行中に変えられる。
- III 要素は添字で識別する。
- IV 各要素は連続した領域に格納される。
- V 各要素は必ずしも連続した領域に格納されなくともよい。

ア I、III、IV

イ I、III、V

ウ I、IV

エ II、IV

解答解説

配列の特徴に関する問題である。

配列の特徴を整理すると次のようになる。

- ① あらかじめ決めた同じ大きさの要素を物理的に連続的に並べたものである。
- ② 一次元から多次元までいろいろな配列が可能である。
- ③ 各要素は添字で参照されるので直接データを指定できる。
- ④ 特定のデータを参照するための処理時間はデータ量に関係なく一定である。

配列の特徴は、大きさはあらかじめ決まっている、要素は添字で識別される、要素は連続した領域に格納されるで、I、III、IVとなる。求める答えはアとなる。

例題演習

三次元の配列を表すとき“ $X(I, J, K)$ ”と表記するが、このとき、Jが表しているものはどれか。

ア 行

イ 面

ウ 要素

エ 列

解答解説

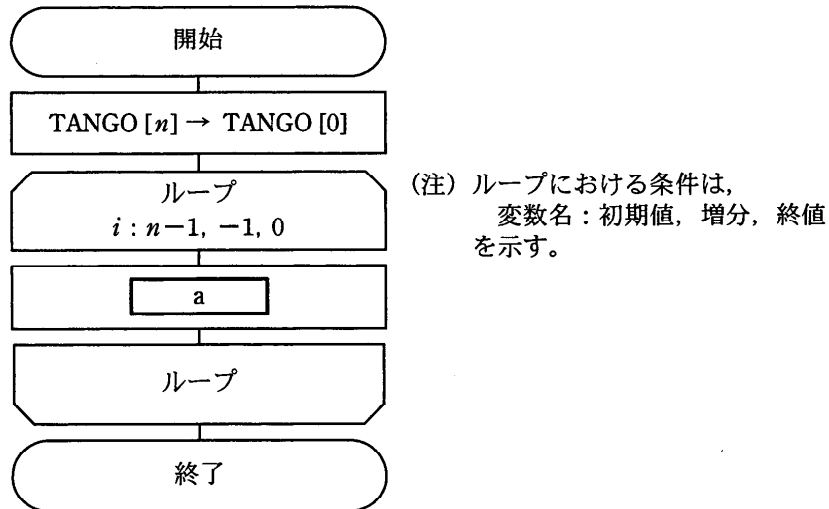
3次元配列の問題である。

3次元の配列は、面、行、列で表現される。

Iは面、Jは行、Kは列を表す。従って、Jは行を表す。求める答えはアとなる。

例題演習

要素番号が0から始まる配列TANGOがある。n個の単語がTANGO[1]からTANGO[n]に入っている。図は、n番目の単語をTANGO[1]に移動するために、TANGO[1]からTANGO[n-1]の単語を順に一つずつ後ろにずらして単語表を再構成する流れ図である。aに入れる処理として、適切なものはどれか。



- ア TANGO[i]→TANGO[i+1]
- イ TANGO[i]→TANGO[n-i]
- ウ TANGO[i+1]→TANGO[n-i]
- エ TANGO[n-i]→TANGO[i]

解答解説

配列の要素の値を後方から一つずつ一方の要素に移動させ、TANGO[1]に空白の要素を作成する流れ図の問題である。

次の手順で操作する。

- ① TANGO[n]→TANGO[0]
- ② iにn-1を格納する。
- ③ TANGO(i)→TANGO(i+1)を実行する。
- ④ i-1→iを実行する。
- ⑤ i>0ならば②に戻る。
- ⑥ i=0になると、TANGO[0]→TANGO[1]を実行する。

要素の移動は、n-1番目の要素の値をn番目に移動し、次にiのディクリメントを行い、n-2番目の要素をn-1番目の要素に移動させる。i番目の要素をi+1番目に移動し、順次、ディクリメントしながら、最後に0番目の要素を1番目の要素に移動させて終了する。従って、aに入るのはTANGO[i]→TANGO[i+1]となる。求める答えはアとなる。

例題演習

10行10列の2次元配列aを、次のようにメモリ上の連続した領域へ行方向に格納するとき、a[5, 6]が格納される場所の番地はどれか。ここで、番地は10進数表示とする。

ア	145	番地	
イ	185	100	
ウ	190	101	---- a[1, 1] ----
エ	208	102	
		103	---- a[1, 2] ----

解答解説

2次元配列を1次元配列に換算する問題である。

a[i, j]の格納されるアドレスは次の式から計算する。

$$100 + 2 \times \{(j - 1) + 10 \times (i - 1)\}$$

上記の式を用いてa[5, 6]を計算すると、次のようになる。

$$100 + 2 \times (5 + 40) = 100 + 90 = 190$$

求める答えはウとなる。

例題演習

長さm、nの文字列を格納した配列X、Yがある。図はこれらの文字列をこの順に連結した文字列を配列Zに格納する処理の流れ図である。a、bに入れる処理として、正しいものはどれか。ここで、1文字が一つの配列要素に格納されるものとする。

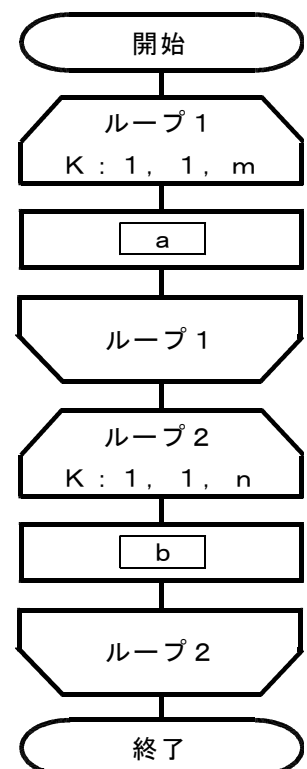
	a	b
ア	X(k) → Z(k)	Y(k) → Z(m + k)
イ	X(k) → Z(k)	Y(k) → Z(n + k)
ウ	Y(k) → Z(k)	X(k) → Z(m + k)
エ	Y(k) → Z(k)	X(k) → Z(n + k)

解答解説

2つの配列を結合する流れ図に関する問題である。

結合する手順は次の通りである。

- ① 最初に大きさmの配列X(k)を配列Zに格納する。
- ② 配列Xを格納後の配列Zの大きさはmである。
- ③ その次に大きさnの配列Y(k)を配列Zに格納する。



④ 配列 Y は $Z(m+1)$ から $Z(m+n)$ の要素に格納される。

⑤ 配列 X、Y を格納後の配列 Z の大きさは $m+n$ となる。

ループ 1 は、配列 X から配列 Z への文字列の格納処理である。従って、配列 X の長さは m であるから、 $X(k) \rightarrow Z(k)$ となる。

ループ 2 は、配列 Y から配列 Z の $m+1$ 番目の要素から n 個の要素を配列 Z に格納し、配列の要素番号 $m+n$ までの要素に格納する。即ち、 $Y(k) \rightarrow Z(m+k)$ となる。

a、b の組み合わせは A となる。求める答えは A となる。

例題演習

次の探索方法のうちで番兵が有効なものはどれか。

ア 2 分探索

イ 線形探索

ウ ハッシュ探索

エ 幅優先探索

解答解説

線形探索の番兵に関する問題である。

番兵が有効なのは線形探索のように、並びの先頭から順次探索していく場合である。

番兵は探し求める値を配列の最後に付加して、配列の大きさ N の範囲内に求める値が見つければ探索成功、 $N+1$ で見つかりと探索不成功の判定をする線形探索法である。従って、番兵が有効となるのは線形探索である。求める答えはイとなる。

例題演習

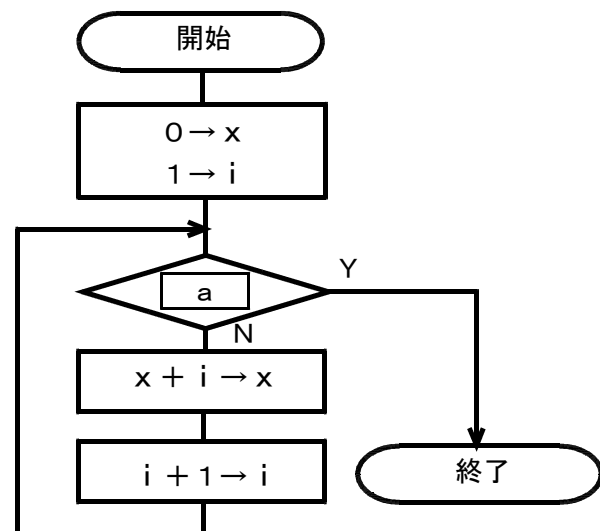
流れ図は、1 から N ($N \geq 1$) までの整数の総和 $(1 + 2 + \dots + N)$ を求め、結果を変数 x に入れるアルゴリズムを示している。流れ図中の a に当てはまる式はどれか。

ア $i = N$

イ $i < N$

ウ $i > N$

エ $x > N$



解答解説

流れ図に関する問題である。

流れ図は $1 \sim N$ までの整数の総和を求める問題である。

アの $i = N$ 、イの $i < N$ では、 N が加算されていない。

エの $x > N$ では、 $N = 3$ で真となり終了する。

a は終了条件の判定であるから $i > N$ となる。求める答えはウとなる。

① リストの定義と特徴

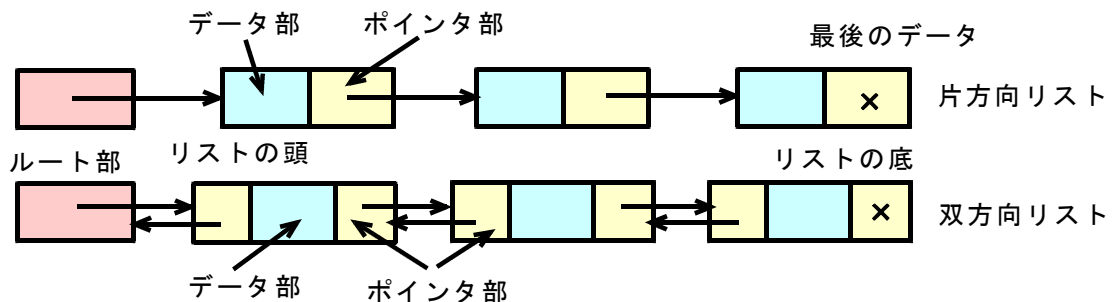
① リストの定義と構成

リストはルート部を先頭にして、各データがポインタ部で結合された一連のデータ群である。各データは物理的に連続に配置されている必要がない。

データはデータ部とポインタ部で構成され、データはデータ部に格納され、ポインタ部には次のデータのアドレスが格納されている。各データはポインタを利用して、順次つながれた構造になっている。リストの最後のデータのポインタ部には、最後のデータであることを示す記号が格納される。

② 片方向リストと双方向リスト

片方向リストは一方にしか辿ることができないリストであり、双方向リストは両方向に辿ることのできるリストである。



② リストを利用したデータ探索

① リストの探索法

リストの探索は、ルート部から開始し各データのポインタ部を用いて、順次、次のデータのアドレスを求め、探索データと各要素の内容を比較する。探索データと各要素の内容を比較して一致すれば探索成功、最後の要素まで比較して不一致ならば探索不成功になる。

ポインタを利用して探索するので線形探索のみ可能であり、直接探索はできない。片方向リストの場合は前方からの探索のみであるが、双方向リストの場合、前方後方いずれの方向にも探索可能である。

② 片方向リストの探索手順

⑦ ルート部から最初のデータのアドレスを知る。

- ① 探索データとデータ部の内容を比較する。
- ㊦ (探索データ=データ部の内容)ならば探索成功、処理を終了する。
- ㊧ (探索データ≠データ部の内容)ならば探索不成功、ポインタ部の情報から次のデータのアドレスを知り、①に戻る。
- ㊨ (探索データ≠最後のデータ部の内容)ならば、探索データが存在しないため、処理を終了する。

③ リストのデータ更新と削除

㊰ リストの更新

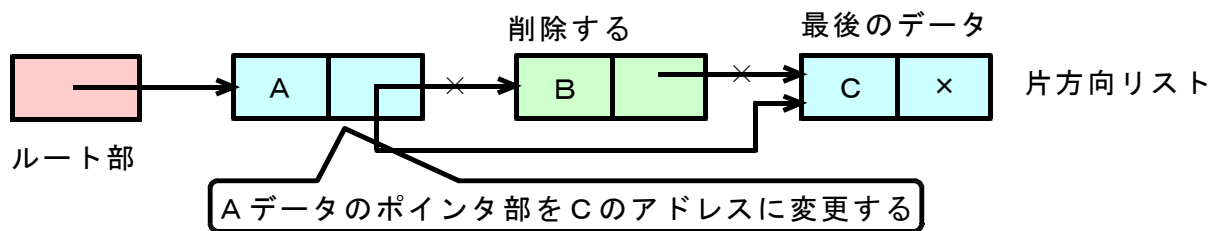
リストの更新は更新すべきデータを線形探索で求め、その内容を書き換える。

㊱ リストの削除

リストの削除は削除するデータの直前のデータのポインタ部を修正することによって行う。

㊲ リストの削除手順

- ㊴ メモリ上にアドレスを格納する領域Qを確保し、ルート部の情報から最初のデータを読む。
- ① 最初のデータが削除するデータならば、最初のデータのポインタ部の情報をルート部に格納し、処理を終了する。
- ㊵ 探索したデータが削除するデータでないならば、探索したデータのアドレスをメモリ領域Qに格納し、探索したデータのポインタ部の情報から次の探索するデータのアドレスを知る。
- ㊶ 探索したデータが削除するデータの場合、削除するデータのポインタ部の情報を用いてメモリ領域Qのデータのポインタ部の内容を修正し、処理を終了する。
- ㊷ 最後のデータになるまで、㊵に戻り処理を繰り返す。
- ㊸ 最後のデータに達しても探索するデータでない場合、処理を終了する。



④ リストのデータ挿入

① リストの挿入

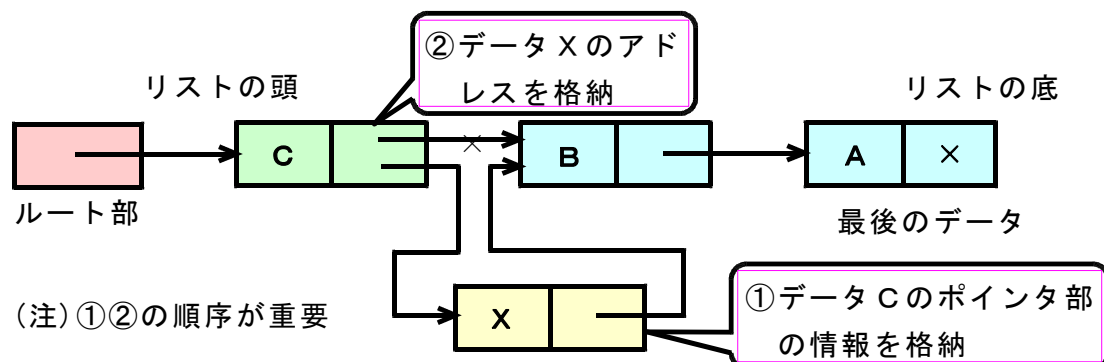
リストの挿入は、挿入するデータの直前のデータのポインタ部の書き換えと挿入するデータのポインタ部への書き込みによって行われる。

② 特定データの直後に挿入

挿入位置を線形探索で決定し、挿入位置のデータのポインタ部の書き換えと新しく挿入するデータのポインタ部への情報の書き込みによって行う。挿入位置のデータのポインタ部の情報を挿入するデータのポインタ部に格納し、挿入位置のデータのポインタ部に新しく挿入するデータのアドレスを格納する。

③ 特定データの直後に挿入する手順

データ C の直後にデータ X を挿入する



㊦ メモリ上のデータ列の先頭位置にデータを挿入する場合、ルート部の情報を挿入するデータのポインタ部に格納し、ルート部の内容を挿入するデータのアドレスに書き換える。

① 最初のデータが挿入位置ならば、挿入位置のデータのポインタ部の情報を挿入するデータのポインタ部に格納し、挿入するデータのアドレスを挿入位置のデー

タのポインタ部に格納し、処理を終了する。

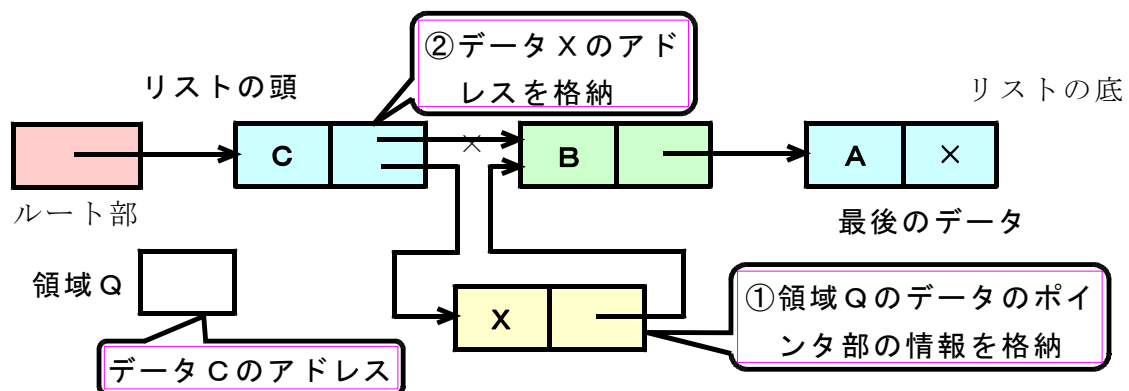
- ㉔ 探索したデータが挿入位置でないならば、探索したデータのポインタ部の情報から次の探索するデータのアドレスを知る。
- ㉕ 探索したデータが挿入位置のデータの場合、挿入位置のデータのポインタ部の情報を挿入するデータのポインタ部に格納し、挿入するデータのアドレスを挿入位置のデータのポインタ部に格納し、処理を終了する。
- ㉖ 最後のデータになるまで、㉔に戻り処理を繰り返す。
- ㉗ 最後のデータに達しても挿入位置がない場合、処理を終了する。

㉘ 特定データの直前に挿入

挿入位置を線形探索で決定し、挿入位置の直前のデータのポインタ部の書き換えと新しく挿入するデータのポインタ部への情報の書き込みによって行う。線形探索で挿入位置を探索する場合、探索データの直前のデータのアドレスを特定の記憶領域に格納する。探索直前のデータのアドレスを記憶していないと、挿入直前のデータのポインタ部を書き換えることができない。挿入位置の直前のデータのポインタ部の情報を挿入するデータのポインタ部に格納し、挿入位置の直前のデータのポインタ部に新しく挿入するデータのアドレスを格納する。

㉙ 特定データの直前に挿入する手順

データ B の直前にデータ X を挿入する



- ㉚ アドレスを格納する領域 Q を確保し、ルート部から最初のデータを読む。
- ㉛ 最初のデータが挿入位置の場合、ルート部の情報を挿入するデータのポインタ部に格納し、挿入するデータのアドレスをルート部に格納し、処理を終了する。

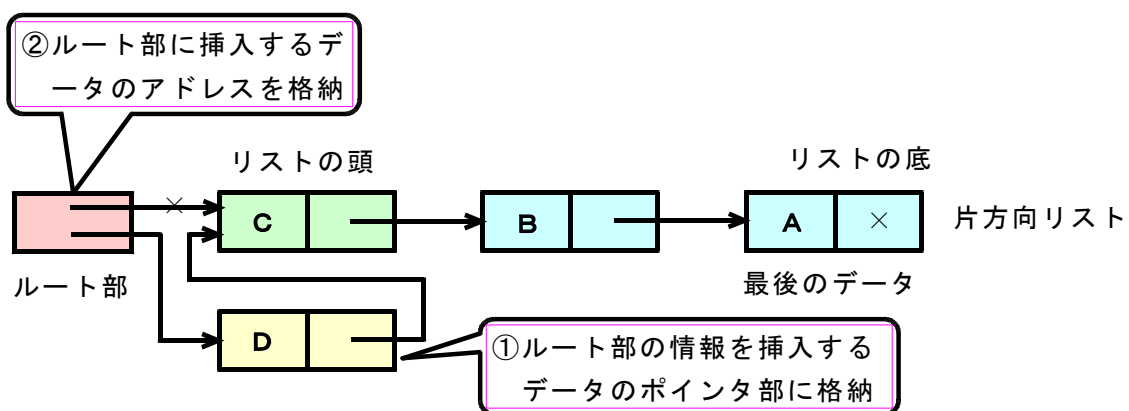
㉔ 探索したデータが挿入位置でない場合、処理㉑に移り、挿入位置の場合、処理㉕に移る。

㉑ 最後のデータでなければ、探索したデータのアドレスを領域Qに格納し、探索したデータのポインタ部の情報から次に探索するデータのアドレスを知り、処理㉔に戻り、最後のデータならば、処理を終了する。

㉕ 領域Qのデータのポインタ部の情報を挿入するデータのポインタ部に格納し、挿入するデータのアドレスを領域Qのデータのポインタ部に格納し、処理を終了する。

㉖ リストの先頭部に挿入

次の図はリストの先頭位置にデータを挿入する場合の状態を示したものである。ルート部の操作によって挿入が可能となる。



㉗ 初期化操作

初期化は最初の要素を指すルート部にリストの終端マークを設定すればよい。

例題演習

次のうちリスト構造の特徴を表している適切な組合せはどれか。

- I データ全体の大きさはあらかじめ決まっている。
- II データ全体の大きさは実行中にえられる。
- III 各要素にはデータ部とポインタ部がある。
- IV 各要素は連続した領域に格納されている。
- V 各要素は必ずしも連続した領域に格納されなくともよい。

ア I、III

イ I、III、IV

ウ II、III、IV

エ II、III、V

解答解説

リストの特徴に関する問題である。

リストの特徴をあげると次のようになる。

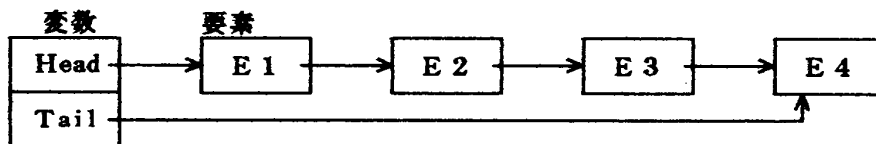
- ① 各要素はデータ部とポインタ部で構成され、各データはポインタでつながれている。ポインタで連結されているので物理的に順番に並んでる必要がない。
- ② 連結された順番でしか探索することができない。線形探索のみである。
- ③ 実行中に大きさを変えることができ、大きさが前もってわかっている必要がない。
- ④ 並べ替えや挿入削除が容易である。

データ全体の大きさは実行中に変えられる、各要素にデータ部とポインタ部がある、各要素は必ずしも連続した領域に格納されなくともよい等の特徴が該当する。従って、Ⅱ、Ⅲ、Ⅴがリストの特徴となる。求める答えはエとなる。

Ⅰ、Ⅳは配列の特徴である。

例題演習

図のような構造をもった線形リストについて、正しい記述はどれか。



- ア 要素の削除に要する処理量は、先頭と最後尾とでほぼ同じである。
- イ 要素の追加と取出し（読出し後削除）を最後尾で行うスタックとしての利用に適している。
- ウ 要素の追加に要する処理量は、先頭と最後尾とでほぼ同じである。
- エ 要素の追加は先頭に、取出し（読出し後削除）は最後尾からとするFIFO(First-in First-out)のキューとしての利用に適している。

解答解説

リストに関する問題である。

アの場合、E1の削除はHeadの内容をE2のアドレスに修正するのみであるが、E4の削除はE3のアドレスを先頭から順次求めて、Tailの修正をする必要があるため両者の処理量は同じではない。

イの場合、E4の追加はTailを操作することで可能であるが、E4の削除は、E4を削除後、E3に設定するために先頭のE1から順次探索する必要があり、スタックのような1変数の処理ではできない。

ウの要素の追加は、先頭への追加は新しい要素にE1のアドレスを記入し、Headに新しい要素のアドレスを格納すればよい。最後尾への追加は追加する要素にE4のポインタの内容を複製し、E4のポインタにTailのアドレスを格納し、Tailに新しい要素のアドレスを格納する。従って、先頭から線形探索する必要がなく、両者の処理量はほぼ同じになる。求める答えはウとなる。

エの待ち行列では、リストの構造の先頭から最後尾に繋がっており先頭から取り出されて最

後尾に追加される構造になる。図の場合は、先頭に追加されて最後尾から取り出されるため最後尾の次の要素の設定が待ち行列のような仕組みにならない。従って、待ち行列のfront、rearの様な条件にならない。

例題演習

図は単方向リストを表している。“東京”がリストの先頭であり、そのポインタには次のデータのアドレスが入っている。また、“名古屋”はリストの最後であり、そのポインタには0が入っている。アドレス150に置かれた“静岡”を、“熱海”と“浜松”の間に挿入する処理として正しいものはどれか。

先頭へのポインタ	アドレス	データ	ポインタ
10	10	東京	50
	30	名古屋	0
	50	新横浜	90
	70	浜松	30
	90	熱海	70
	150	静岡	

- ア 静岡のポインタを50とし、浜松のポインタを150とする。
- イ 静岡のポインタを70とし、熱海のポインタを150とする。
- ウ 静岡のポインタを90とし、浜松のポインタを150とする。
- エ 静岡のポインタを150とし、熱海のポインタを90とする。

解答解説

リスト構造のポインタ操作に関する問題である。

問題の図の内容は、東京→新横浜→熱海→浜松→名古屋 の順を示している。

熱海と浜松の間に静岡を挿入すると、東京→新横浜→熱海→**静岡**→浜松→名古屋 となり、熱海のポインタを静岡のアドレス150に変更し、静岡のポインタを浜松のアドレス70に変更すればよい。求める答えはイとなる。

例題演習

多数のデータが単方向リスト構造で格納されている。このリスト構造には、先頭ポインタとは別に、末尾のデータを指し示す末尾ポインタがある。次の操作のうち、ポインタを参照する回数が最も多いものはどれか。

- ア リストの先頭にデータを挿入する。
- イ リストの先頭のデータを削除する。
- ウ リストの末尾にデータを挿入する。
- エ リストの末尾のデータを削除する。

解答解説

リスト構造に関する問題である。

アのリストの先頭へのデータの挿入は、現在の先頭ポインタの値を挿入するデータのポインタ部に格納し、挿入データのアドレスを先頭ポインタに格納すれば処理できる。参照するポインタの回数は2回である。

イのリストの先頭のデータの削除は、先頭のデータまで探索し、先頭のデータのポインタ部の値を先頭ポインタに格納することで処理できる。参照するポインタの回数は3回である。

ウのリストの末尾へのデータの挿入は、末尾ポインタを利用して、末尾データのポインタ部と末尾ポインタに挿入するデータのアドレスを格納し、挿入データのポインタ部の末尾処理を行う。参照するポインタの回数は2回である。

エのリストの末尾データの削除は、先頭ポインタから順次探索し、末尾データの一つ前のデータのポインタ部の求め、末尾ポインタの終端記号を一つ前のデータのポインタ部に格納する。データ個数が3個の場合、ポインタ参照回数は4回となり、データ個数が4以上になるとポインタ参照回数は5以上になる。末尾データの一つ前のデータの探索が必要であり、ポインタを参照する回数が少なくとも4回以上必要となり、最も多くなる。求める答えはエとなる。

例題演習

次のような双方向のポインタをもつリスト構造のデータがある。社員Gを社員Aと社員Kの間に追加する場合、追加後の表のポインタa～fのうち、追加前と比べて値が変わるのは何か所か。

追加前

アドレス	社員名	次ポインタ	前ポインタ
100	社員A	300	0
200	社員T	0	300
300	社員K	200	100

追加後

アドレス	社員名	次ポインタ	前ポインタ
100	社員A	a	b
200	社員T	c	d
300	社員K	e	f
400	社員G	x	y

ア 1 イ 2 ウ 3 エ 4

解答解説

リストの挿入に関する問題である。

最初の社員の並びは 社員A→社員K→社員T であり、社員Aと社員Kの間に社員Gを挿入した場合の並びは 社員A→社員G→社員K→社員T となる。

アドレス、社員名、ポインタの関係は表のようになる。

a～fの間で変化したのはaの300→400とfの100→400の2カ所である。

求める答えはイとなる。

アドレス	社員名	次ポインタ	前ポインタ
100	社員 A	400	0
200	社員 T	0	300
300	社員 K	200	400
400	社員 G	300	100

例題演習

リストは、配列で実現する場合とポインタで実現する場合とがある。リストを配列で実現した場合の特徴として、適切なものはどれか。

- ア リストにある実際の要素数にかかわらず、リストの最大長に対応した領域を確保し、実際には使用されない領域が発生する可能性がある。
- イ リストにある実際の要素数にかかわらず、リストへの挿入と削除は一定時間で行うことができる。
- ウ リストの中間要素を参照するには、リストの先頭から順番に要素をたどっていくので、要素数に比例した時間が必要となる。
- エ リストの要素を格納する領域の他に、次の要素を指し示すための領域が別途必要となる。

解答解説

配列を利用したリスト構造の特徴に関する問題である。

アの配列の大きさはリストの最大長で決めるため、実際のリストの大きさでは使用されない領域が発生する。求める答えはアとなる。

イの挿入、削除に要する時間は、配列サイズが大きくなるほど、また、操作位置が先頭に近くなるほど長時間になる。

ウの中間参照に必要な時間は、インデックス、添字が利用できるため、高速となり、リストの要素数には関係しない。比例しない。

エの配列を利用すると、連続した領域に格納されるためポインタに相当する領域は必要ない。

① スタックの定義と特徴

① スタックの定義

スタックは、データの挿入と削除が一方の端でのみ行われる。挿入や削除が行われる一方の端を頂上、もう一方の端を底という。挿入をプッシュ (PUSH)、削除をポップ (POP) という。

② スタックポインタ

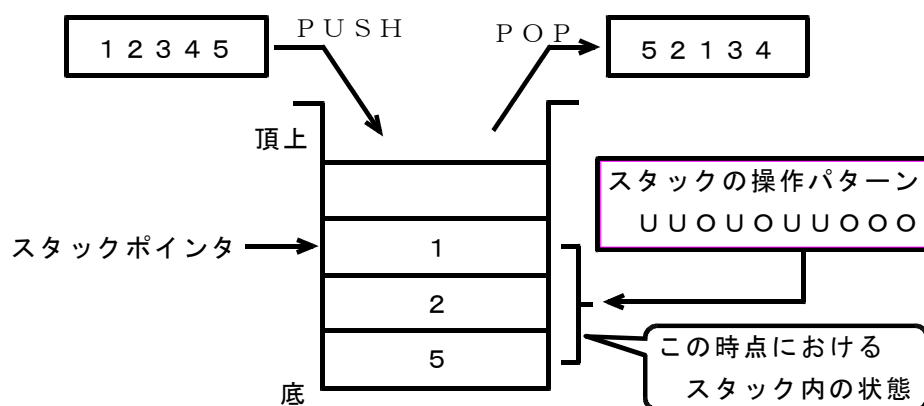
スタックはスタックポインタという一つの変数によってデータを管理する。スタックポインタはスタック内の一番上のデータのアドレスを示す。データが挿入されるとスタックポインタは1増加し、データが削除されると1減少する。

③ スタックの特徴

- ㊦ 要素の挿入や削除はスタックの先頭の領域でのみ行われる。
- ㊧ 後入れ先出しの特徴をもった構造である。
- ㊨ 要素の挿入をPUSH、取り出しをPOPという。
- ㊩ プログラムからサブルーチンや関数などを呼び出すときにスタックを使用し、割り込み処理やサブルーチンの戻り番地の管理に利用される。

② スタックの操作

① スタックに関する操作と状態

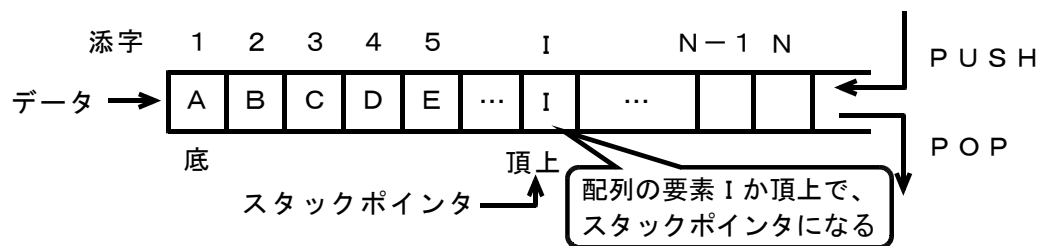


- ㊦ プッシュ前のデータの並び
- ㊧ ポップ後のデータの並び

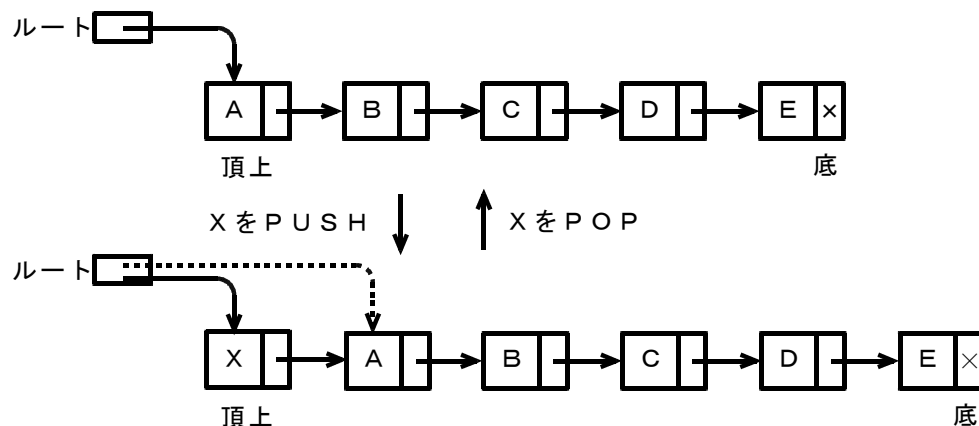
- ㊤ プッシュ・ポップの操作パターン
- ㊥ プッシュ・ポップが行われた直後のスタック内のデータの状態
- ㊦ スタックポインタの値

⑥ 配列を利用したスタック

大きさNの配列 $TBL(I)$ を用いてスタックを実現するとき、 $TBL(1)$ がスタックの底を表し、 $TBL(N)$ がスタックの頂上になる。スタックポインタはスタックの頂上のアドレスを示し、配列の添字が用いられる。スタックにデータを格納するとき、スタックのオーバーフローをチェックし、スタックからデータを取り出すとき、スタックが空でないことをチェックする。



⑦ リストを利用したスタック

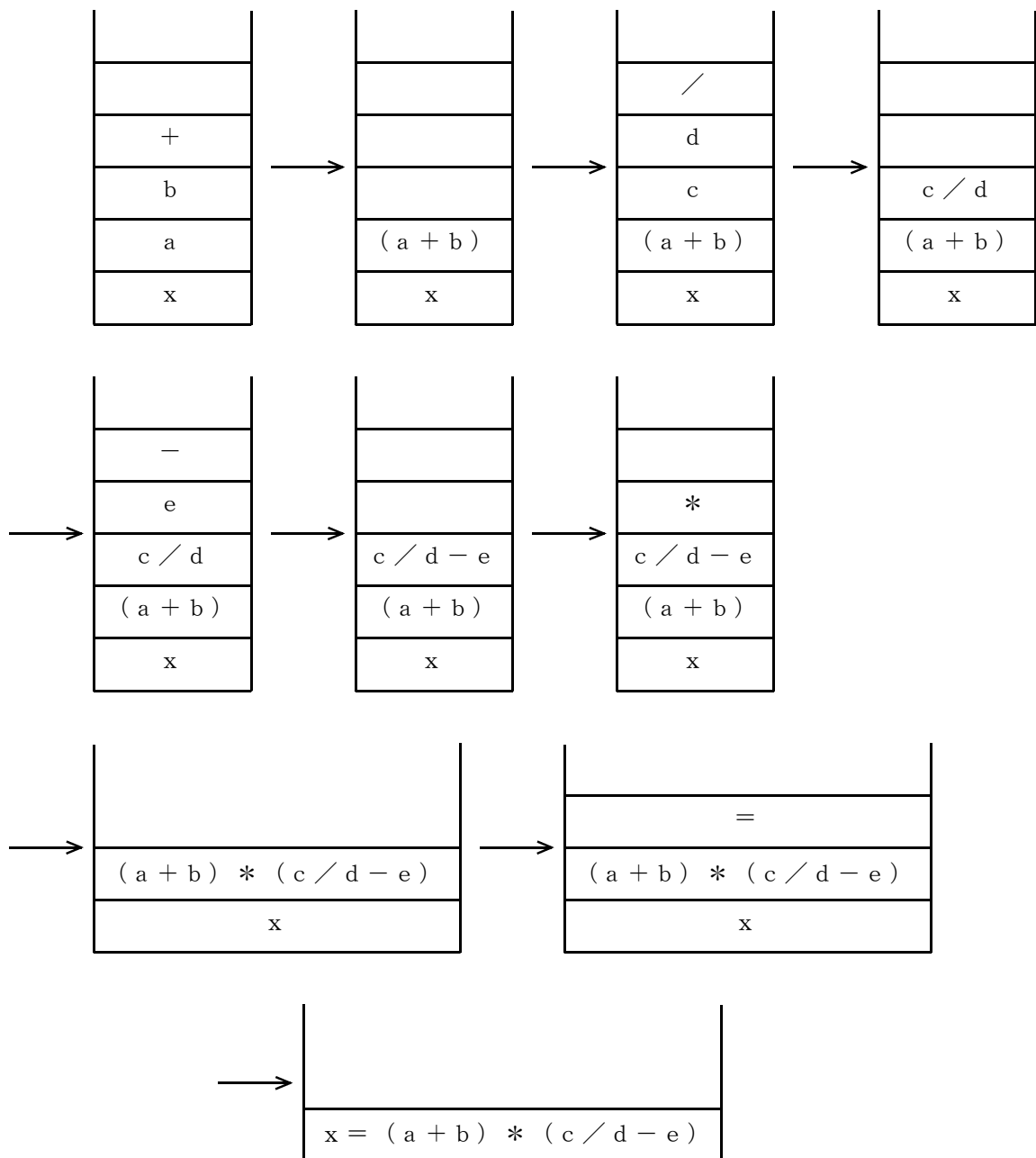


リストの最初のデータを頂上とし、リストの最後のデータを底とする。スタックへのデータの格納と取り出しはリストの先頭から行う。データの格納はリストの先頭への挿入であり、データの取り出しはリストの先頭のデータを削除することである。データを取り出すとき、リストが空であるかどうかはルートの内容が終端記号であるかどうかでわかる。

③ スタックの応用

㊐ 四則演算の具体例

$x = (a + b) * (c / d - e)$ を逆ポーランド記法 $x \ a \ b \ + \ c \ d \ / \ e \ - \ * \ =$ に変換し、スタックを使用して計算する場合の計算途中のスタック内の状態の変化を次に示す。



⑥ スタックを利用した四則演算の手順

スタックを使用した四則演算の手順を次に示す。

- ㊦ スタックポインタを初期化する。
- ㊩ 先頭から=が現れるまで、㊦、㊨、㊧の処理を繰り返す。
- ㊦ 変数または数値ならばスタックにPUSHする。
- ㊧ 演算子が現れると、次の処理を行う。
 - ① スタックから2つの要素を連続してPOPする。
 - ② 取り出した2つの要素を(後で取り出した要素)演算子(先に取り出した要素)の順に処理する。

- ③ 演算結果をスタックにPUSHする。
- ④ =が現れると、スタックの内容をPOPし、処理を終了する。

④ 待ち行列の定義と特徴

④a 待ち行列の定義

待ち行列は、一方の端で挿入が行われ、他方の端で削除が行われるものである。新しいデータは最後のデータの後ろに挿入され、最初の古いデータが削除の対象になる。

待ち行列は2つの変数によって管理される。frontは先頭のデータのアドレスであり、rearは末尾の次のアドレスである。

利用例として、多重プログラミングの実行待ちプログラムの待ち行列がある。

④b 待ち行列の特徴

- ㊦ 要素の挿入が一方の端で行われ、取り出しが他方の端で行われるデータ構造である。
- ㊧ 待ち行列は先入先出、F I F Oの特徴をもつ。
- ㊨ front、rearの二つの変数で管理される。
- ㊩ 使用例として、サービスを受けるために待っている行列がある。

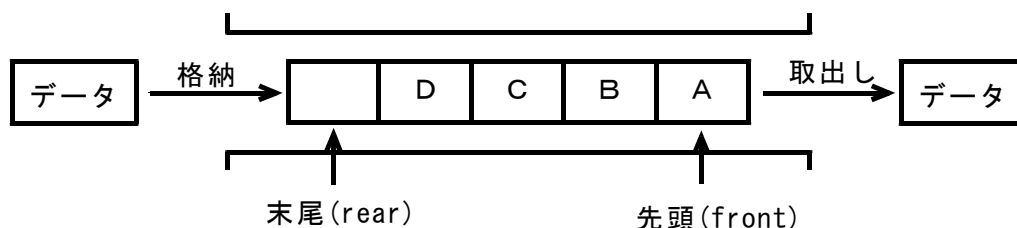
⑤ 待ち行列の操作

⑤a 配列を利用した待ち行列

配列を利用して、待ち行列の先頭を示す変数frontと末尾を示す変数rearを定義し、この2つの変数に挟まれる部分を待ち行列と見なす。

変数rearは配列の最後の要素の次の要素の添字を示す。変数rearと変数frontの差がデータの個数を示す。変数rearと変数frontが一致したとき配列が空になる。

⑤b 配列による待ち行列の格納・取出しの操作



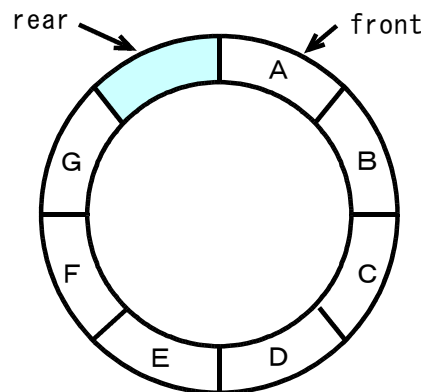
- ㊦ 配列の添字の小さい要素から順にデータを格納する。

- ① データを1つ格納すると変数rearを1増加させる。
- ② データを1つ取出すと変数frontを1増加させる。

㉓ 配列を利用した待ち行列の問題点

配列を利用した待ち行列を考えると、無限大の長さの待ち行列が必要になる。メモリ空間内に無限の配列を考えることは不可能であり、リングバッファの考え方を利用し、擬似無限配列を作成し利用する。

㉔ リングバッファを利用した待ち行列



リングバッファは配列を環状にし、配列の最後の次は配列の先頭を指すように操作する。リングバッファを使用することによって、有限のメモリ領域で無限の待ち行列を実現することができる。

リングバッファが満杯になったとき、先頭＝末尾となり、待ち行列の空の状態 $rear = front$ に等しくなる。満杯と空の状態が同じ条件になり、論理的に矛盾する。その対策として、待ち行列が空である変数を設けたり、最大個数－1の個数しか格納できない規則を設けたりして、この矛盾を解決する。

例題演習

A, B, C, Dの順に到着するデータに対して、一つのスタックだけを用いて出力可能なデータ列はどれか。

- ア A, D, B, C
- ウ C, B, D, A

- イ B, D, A, C
- エ D, C, A, B

例題解説

スタックに関する問題である。

アの場合、U O U U U Oの操作でADは可能であるが、次にBをPOPすることができない。

イの場合、U U O U U Oの操作でB Dは可能であるが、次にAをP O Pすることができない。
 ウの場合、U U U O O U O Oの操作でC B D Aが可能である。求める答えはウとなる。
 エの場合、U U U U O Oの操作でD Cは可能であるが、次にAをP O Pすることができない。

例題演習

スタックに関する記述として、適切なものはどれか。

- ア 最後に格納したデータを最初に取り出すことができる。
- イ 最初に格納したデータを最初に取り出すことができる。
- ウ 探索キーからアドレスに変換することによって、データを取り出すことができる。
- エ 優先順位の高いデータを先に取り出すことができる。

解答解説

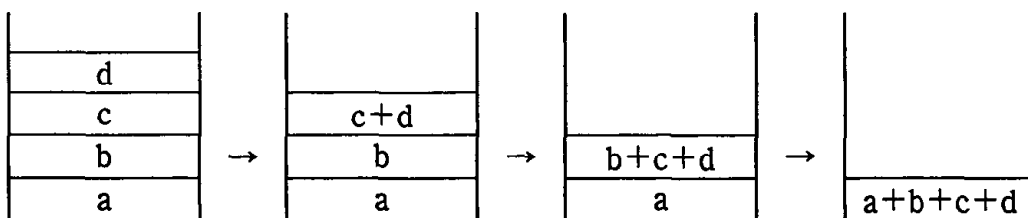
スタックに関する問題である。

スタックはデータの挿入と削除が一方の端でのみ行われる。挿入をプッシュ(P U S H)、削除をポップ(P O P)という。後入先出、L I F Oの特徴をもつ。プログラムからサブルーチンや関数などを呼び出すときにスタックを使用する。

アがスタック、イが待ち行列、ウはハッシュ、エは優先順序別待ち行列であり、求める答えはアとなる。

例題演習

図は、逆ポーランド表記法で書かれたa b c d + + +をスタックで処理するときのスタックの変化の一部を表している。この場合、スタックの深さは最大で4となる。最大のスタックの深さが最も少ない逆ポーランド表記法の式はどれか。



- ア a b + c + d +
- イ a b + c d + +
- ウ a b c + + d +
- エ a b c + d + +

解答解説

スタックの最大深さに関する問題である。

アの場合、a b +で深さ2、(a + b) c +で深さ2、(a + b + c) d +で深さ2、最大深さは2となる。

イの場合、(a + b) c d +で、最大深さは3となる。

ウの場合、a b c +で最大深さ3となる。

エの場合、a b c +で最大深さ3となる。

最大深さが最も小さいのはアの2である。求める答えはアとなる。

例題演習

次の二つのスタック操作を定義する。

PUSH : スタックにデータ（整数n）をプッシュする。

POP : スタックからデータをポップする。

空のスタックに対して、次の順序で操作を行った結果はどれか。

PUSH 1 → PUSH 5 → POP → PUSH 7 → PUSH 6
→ PUSH 4 → POP → POP → PUSH 3

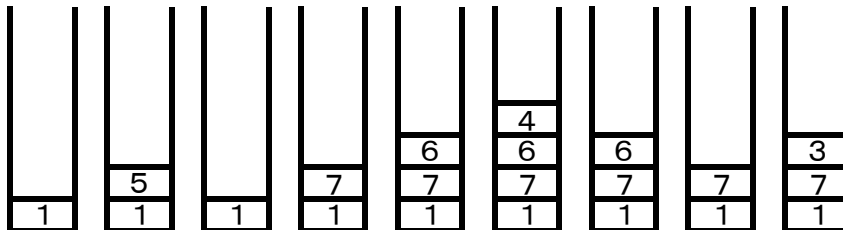
ア	イ	ウ	エ
3	3	3	6
4	4	7	4
5	6	1	3

解答解説

スタックの操作に関する問題である。

示された操作パターンであるPUSH、POPの操作を逐次実行する。

PUSH、POPの操作を実行すると次の図のようになる。スタックの最後の結果はウとなる。求める答えはウとなる。



例題演習

キューに四つの値8、3、6、1がこの順に格納されている。このキューから最初に取り出される値はどれか。

ア 1 イ 3 ウ 6 エ 8

解答解説

キューに関する問題である。

キューの特徴はFIFOであるから、格納した順序に取り出される。格納順は、8 → 3 → 6 → 1であるから、最初に取り出される値は8であり、求める答えはエとなる。

例題演習

十分な大きさの配列 A と初期値が 0 の変数 p に対して、関数 f (x) と g () が次のとおり定義されている。配列 A と変数 p は、関数 f と g だけでアクセス可能である。これらの関数が操作するデータ構造はどれか。

```
function f(x) {  
    p = p + 1  
    A[p] = x  
    return None  
}  
  
function g() {  
    x = A[p]  
    p = p - 1  
    return x  
}
```

ア キュー

イ スタック

ウ ハッシュ

エ ヒープ

解答解説

スタックに関する問題である。

function f() は、ポインタ p をインクリメントし、要素番号 p の配列 A の要素に x を格納する。

function g() は、要素番号 p の配列 A の要素の内容を x に代入して、ポインタ p をデクリメントし、x を返す。関数 f は PUSH の操作であり、関数 g は POP の操作であるから、スタックのデータ構造となる。求める答はイとなる。

例題演習

スタックとキューの二つのデータ構造がある。次の手続きを順に実行した場合、変数 x に代入されるデータはどれか。ここで、データ a をスタックに挿入することを push(a)、スタックからデータを取り出すことを pop()、データ a をキューに挿入することを enq(a)、キューからデータを取り出すことを deq()、とそれぞれ表す。

push(a)、push(b)、enq(pop())、enq(c)、push(d)、push(deq())、x = pop()

ア a

イ b

ウ c

エ d

解答解説

スタックとキューに関する問題である。

問題の操作を順次実行すると、スタック、キュー、x の内容は次のように変化する。

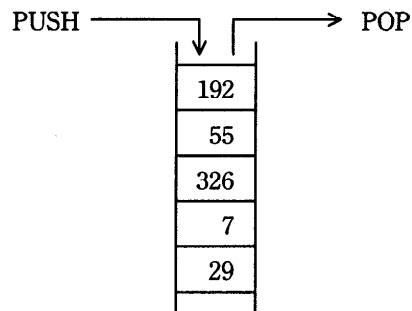
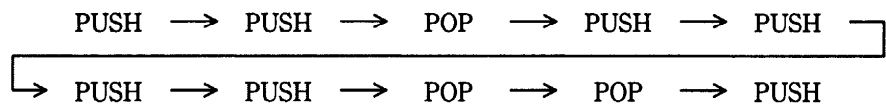
操作順序 1 2 3 4 5 6 7

スタック a → b a → a → a → d a → b d a → d a
 キュー → b → c b → c b → c → c
 x → b
 xに格納されるのはbとなり、求める答えはイとなる。

例題演習

PUSH命令でスタックにデータを入れ、POP命令でスタックからデータを取り出す。動作中のプログラムにおいて、ある状態から次の順で10個の命令を実行したとき、スタックの中のデータは図のようになった。1番目のPUSH命令でスタックに入れたデータはどれか。

- ア 7
- イ 29
- ウ 55
- エ 326



解答解説

スタックに関する問題である。

最後にPUSHであるから、最後に入れたものから何番前のものが最初のものであるかを調べればよい。

- ① PU→PU→POの3回の操作で、スタックの中に操作後のものが1個残っている。
 - ② PU→PU→PU→PU→PO→POの6回の操作で、スタックの中に残るのは2個となり、①のものと合わせると3個となる
 - ③ 最後の操作はPUであるから、操作を開始してから後に4個がスタックに残ったことになる。
- 4番目のものが最初にPUSHしたものであるから7となる。求める答えはアとなる。

例題演習

四つのデータA, B, C, Dがこの順に入っているキューと空のスタックがある。手続pop_enq, deq_pushを使ってキューの中のデータをD, C, B, Aの順に並べ替えるとき、deq_pushの実行回数は最小で何回か。ここで、pop_enqはスタックから取り出したデータをキューに入れる操作であり、deq_pushはキューから取り出したデータをスタックに入れる操作である。

- ア 2
- イ 3
- ウ 4
- エ 5

① 二分木の定義と表現

① 二分木の定義

- ㊦ 空の木は二分木である。
- ㊧ 次のいずれかを満足する節だけからなる木は二分木である。
 - ① 子をもたない木
 - ② 左の子だけもつ木
 - ③ 右の子だけもつ木
 - ④ 左右二つの子をもつ木

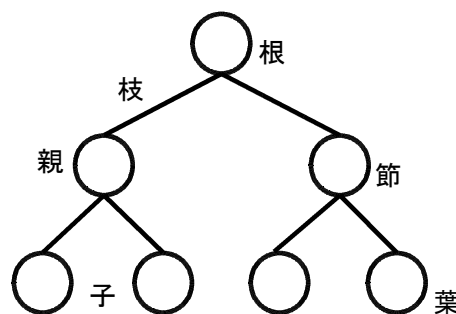
② 部分木

左の子を根とする部分木を左部分木、右の子を根とする部分木を右部分木という。

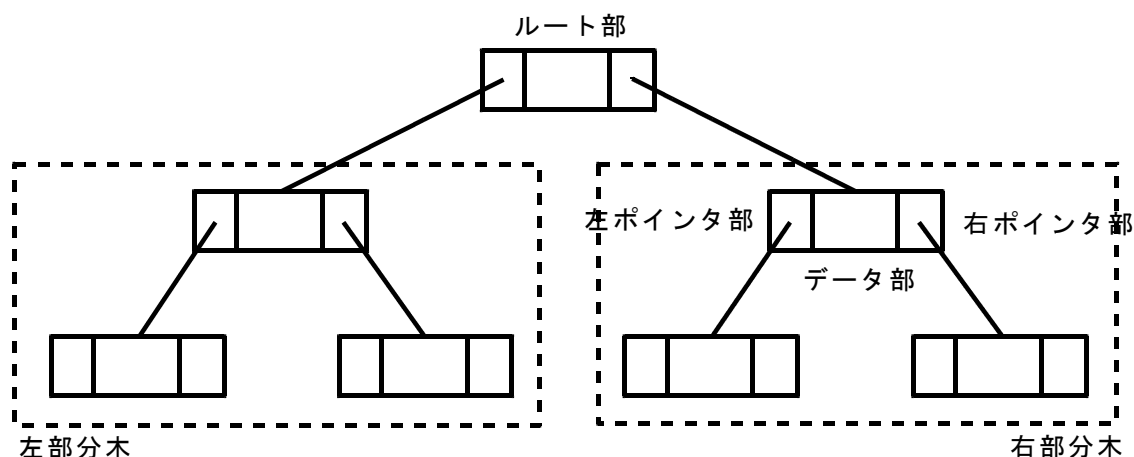
二分木はデータ部とポインタ部で構成される。二分木は最高二つの子をもつので、ポインタ部も二つ必要である。左の子のアドレスを示すポインタ部を左ポインタ、右の子のアドレスを示すポインタ部を右ポインタという。

② 二分木の構成

① 木構造(ツリー構造)の構成要素



- ㊦ ノード(節)はキー値、データが対応する。
- ㊧ ブランチ(枝)はノードとノードを結ぶラインが対応する。
- ㊨ ルート(根)は最上位のノード、親を持たないノードである。
- ㊩ 子はノードの下部に分岐するノード、親のノードの下部のノードが対応する。
- ㊪ 親は分岐元のノード、子の上部のノードが対応する。
- ㊫ リーフ(葉)は最下位のノード、子を持たないノードが対応する。



③ 二分探索木の定義

① 二分探索木の定義

二分探索木は、データ構造の一つである二分木のうち、各ノードの値よりも左の子のノードの値の方が小さく、右の子のノードの値の方が大きくなるように値を挿入したものである。

親と同じ値をどちらのノードに入れるかは任意であるが、毎回異なることがないようにあらかじめ決めておく。

② 二分探索木の条件

- ㊦ 任意の根とその部分木について、左部分木に含まれる各要素は根の要素よりも小さい。
- ㊧ 任意の根とその部分木について、右部分木に含まれる各要素は根の要素よりも大きい。

④ 二分探索木の探索

① 二分探索木を使用した探索

根から枝分かれした二分探索木のデータ構造を利用して、データの探索を行うことができる。二分探索木の条件に従って、各節にデータを置き、根から葉に向かって辿る経路がそのまま探索のアルゴリズムになる。節の値と探索データの値を比較して、探索データの値が小さい場合は左の部分木を、探索データが大きい場合は右の部分木を探索する。

② 二分探索木の探索手順

- ㊦ 探索データをKとする。

- ① 根の値とKを比較する。
- ② 根の値=Kならば、探索成功で処理を終了する。
- ③ 根の値>Kならば、探索データは左部分木にあるので左の節を次の根と考えて①に戻る。
- ④ 根の値<Kならば、探索データは右部分木にあるので右の節を次の根と考えて①に戻る。
- ⑤ 比較対象となる根が存在しなくなると、該当する探索データが存在しないことになり、探索不成功で処理を終了する。

③ 二分探索木の探索時間

探索時間は枝分かれしているレベルの高さに依存する。探索木のレベルが高いほど探索に時間がかかる。二分探索木ではレベルが低いほど効率がよいことになる。

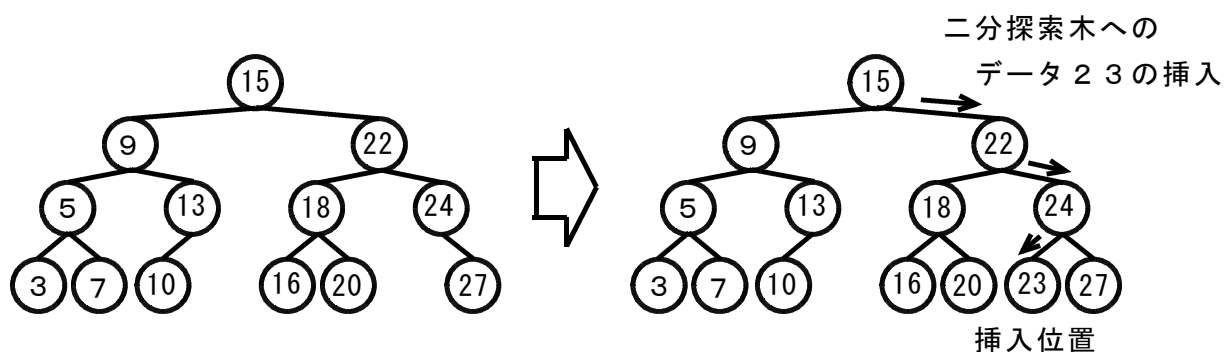
⑤ 二分探索木の挿入

① 挿入位置の探索

挿入位置を見つけるために、挿入するデータを探索データとして二分探索木による探索を実行する。根からはじめて二分探索木を葉に向かって辿り探索する。

② 二分探索木のデータの挿入手順

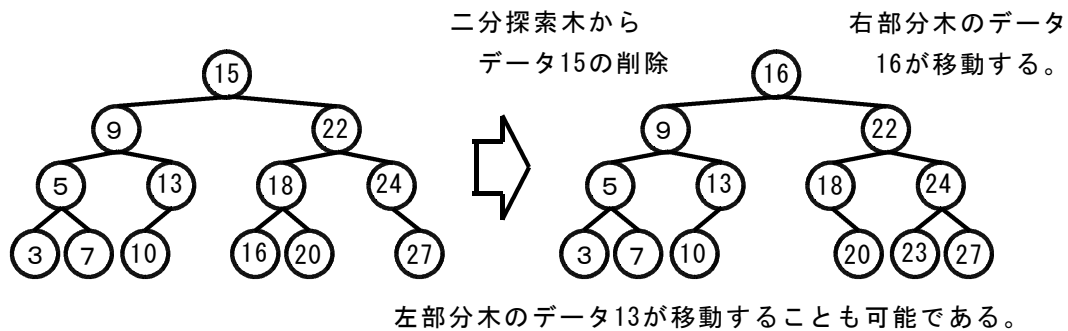
- ⑦ 探索は二分探索木の根から始める。
- ⑧ 探索データの値と節の値を比較する。
 - ① 探索データと節の値が一致すると、挿入データが挿入済みであるので処理を終了する。
 - ② 探索データが小さいと、左の節に移動する。
 - ③ 探索データが大きいと、右の節に移動する。
- ⑨ 探索すべき節または葉が存在しなくなると、その位置に新しいデータを挿入する。探索すべき節が存在すると、⑧にもどる。



⑥ 二分探索木の削除

① データの削除

削除するデータを探索し、削除したデータを節とする右部分木または左部分木を考えて、左部分木の最も大きい値の節かまたは右部分木の最も小さい値の節を削除した節に移動させる。以下同様の処理を繰り返す。



② 二分探索木のデータの削除手順

- ㊦ 二分探索木を用いて探索し、削除すべきデータを見つけ削除する。
- ㊧ 削除した節が子を持たないときは何もしない。
- ㊨ 削除した節が子を持つとき、その子を親の代わりにする。
- ㊩ 削除した節が子を二つ持つ場合は、右部分木に含まれる最小のデータまたは左部分木に含まれる最大のデータを親の代わりとして移動する。
- ㊪ 移動した子が更に子を持つときは、移動した子が削除されたものとして、同様の処理を繰り返す。

例題演習

データ構造の一つである木構造に関する記述として、適切なものはどれか。

- ア 階層の上位から下位に節点をたどり、データを取り出すことができる構造である。
- イ 格納した順序でデータを取り出すことができる構造である。
- ウ 格納した順序とは逆の順序でデータを取り出すことができる構造である。
- エ データ部と一つのポインタ部で構成されるセルをたどることによって、データを取り出すことができる構造である。

解答解説

データ構造に関する問題である。

木構造は節点と枝が木の形をした構成になっている。頂点の節点を根と呼び、末端の節点を葉と呼ぶ。上位側の節点を親、下位側の節点を子、同列の節点を兄弟と呼ぶ。根から特定の節

点まで行く枝の数を深さという。階層の上位から下位に節点を辿ることによって、データを取り出すことができる。

アの上位から下位に節点を辿ることによって、データを取り出す内容は木構造に関するものである。求める答えはアとなる。

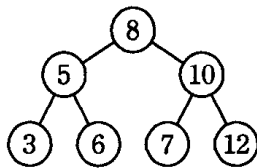
イ、ウは順編成のファイルのデータ構造である。

エは片方向リストのデータ構造である。

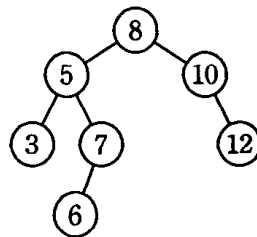
例題演習

空の2分探索木に、8、12、5、3、10、7、6の順にデータを与えたときにできる2分探索木はどれか。

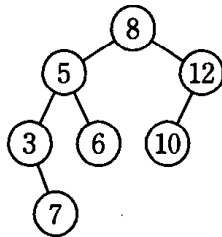
ア



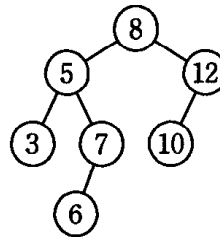
イ



ウ



エ



解答解説

2分探索木に関する問題である。

2分探索木のデータの挿入手順

- ① 探索は二分探索木の根から始める。
- ② 探索データの値と節の値を比較する。

探索データと節の値が一致すると、挿入データが挿入済みであるので処理を終了する。

探索データが小さいと、左の節に移動する。探索データが大きいと、右の節に移動する。

- ③ 探索すべき節または葉が存在しなくなると、その位置に新しいデータを挿入する。探索すべき節が存在すると、②にもどる。

解答群の中のア～エについて、2分探索木の条件、挿入手順の条件を満足するかどうかを検討する。

アの場合、7の位置が8の右部分木になり、探索木の条件を満たさない。

イの場合、8、12、5、3と格納し、10を格納する場合、8の右部分木、12の左部分木の順に格納される。10の格納位置は12の左部分木になる。

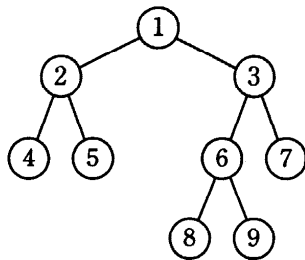
ウの場合、7が5の左部分木になっており、探索木の条件を満たさない。

エの場合、探索木の条件を満足し、挿入条件にも一致する。求める答えはエとなる。

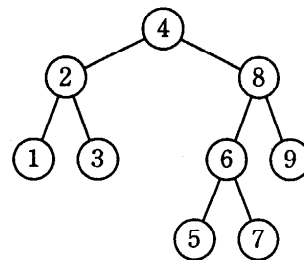
例題演習

2分探索木として適切なものはどれか。ここで、1～9の数字は、各ノード(節)の値を表す。

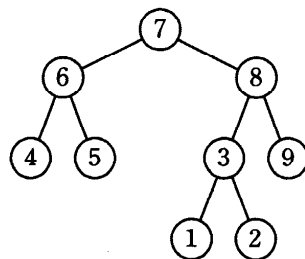
ア



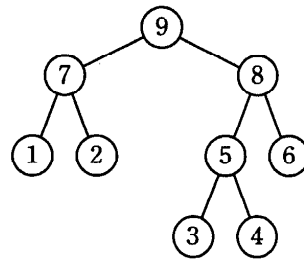
イ



ウ



エ



解答解説

2分探索木に関する問題である。

2分探索木の条件

- ① 任意の根とその部分木について、左部分木に含まれる各要素は根の要素よりも小さい。
- ② 任意の根とその部分木について、右部分木に含まれる各要素は根の要素よりも大きい。

上記の条件を満たす2分木はイである。求める答えはイとなる。

アは、1と2、2と4、3と6、6と8が問題である。

ウは、6と5、3と2が問題である。

エは、7と2、8と6、5と4が問題である。

例題演習

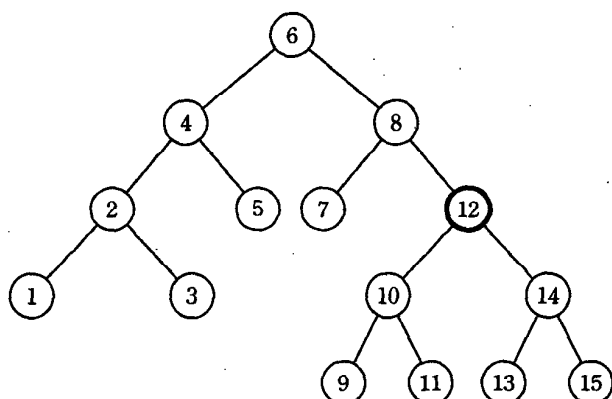
次の2分探索木から要素12を削除したとき、その位置に別の要素を移動するだけで2分探索木を再構成するには、削除された節点の位置にどの要素を移動すればよいか。

ア 9

イ 10

ウ 13

エ 14



解答解説

2分探索木に関する問題である。

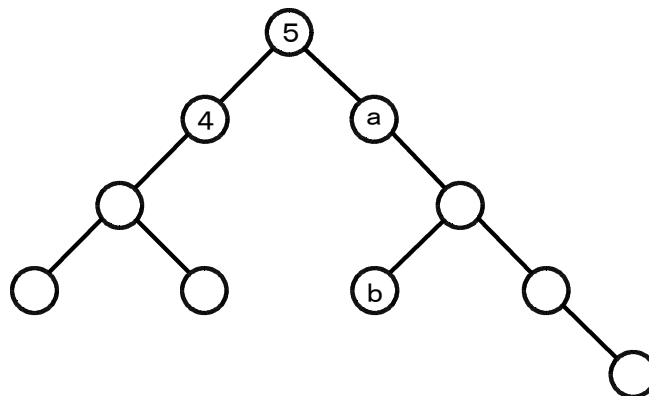
2分探索木の条件

- ① 任意の根とその部分木について、左部分木に含まれる各要素は根の要素よりも小さい。
- ② 任意の根とその部分木について、右部分木に含まれる各要素は根の要素よりも大きい。

12のノードを削除する場合であるから、左部分木の最大値は11、右部分木の最小値は13となり、いずれかが12のノードの部分に移動する。解答群の中には13があるから、求める答えはウとなる。

例題演習

10個の節（ノード）からなる次の2分木の各節に、1から10までの値を一意に対応するように割り振ったとき、節a、bの値の組合せはどれになるか。ここで、各節に割り振る値は、左の子及びその子孫に割り振る値より大きく、右の子及びその子孫に割り振る値より小さくする。



ア a = 6、b = 7

イ a = 6、b = 8

ウ a = 7、b = 8

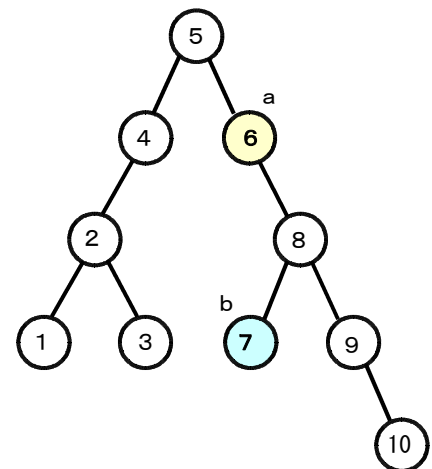
エ a = 7、b = 9

解答解説

探索2分木に関する問題である。

与えられた問題の2分木を完成すると図のようになる。

aは黄色の箇所、bは青色の箇所であるから、A = 6、b = 7となり、求める答えはアとなる。

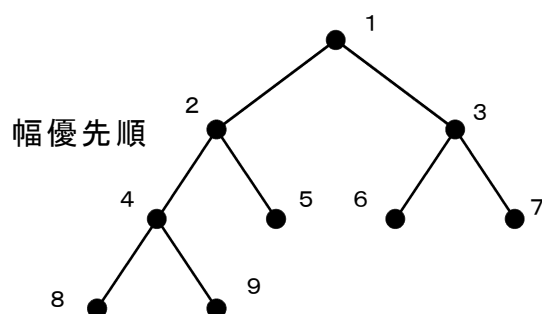
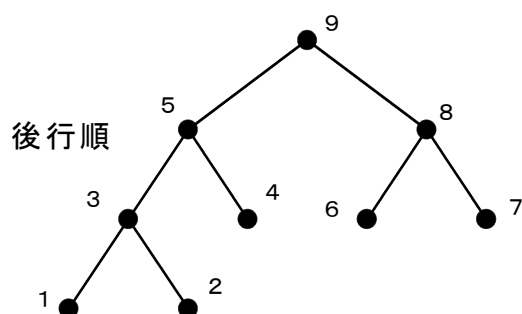
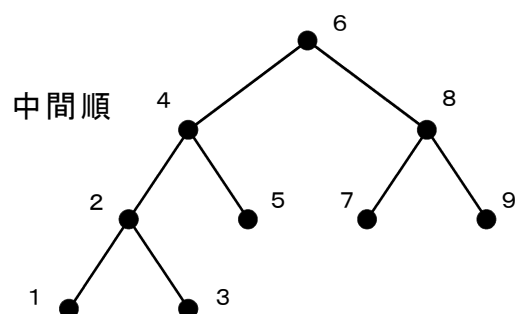
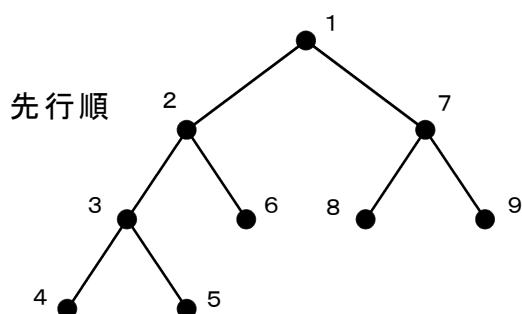


① 二分木の巡回法

① 深さ優先順

深さ優先順の巡回法は、根から始めて左の子からかつ葉の方向に向かって、左の子から右の子の順に巡回を行う方法である。木の深さの方向に対して巡回を繰り返すことになる。

② 巡回の順番を決める方法



㊦ 先行順

節点に来たときに先に順位を割り当てる方法である。

① 中間順

節点に来たときにその左の子を根とする部分木を巡回し終わってから、戻ってきたときに順位を割り当てる方法である。節点の左部分木に順位を割り当てた後に、その節点に順位を割り当てる。

㊧ 後行順

節点に来たときに自分を根とする部分木を巡回し終わってから、その節点に順位を割り当てる方法である。節点の左部分木、右部分木に順位を割り当てた後に、その節点に順位を割り当てる。

㉓ 幅優先順

幅優先順の巡回法は、根から始まって、深さの浅い方からかつ左から巡回する方法である。木の幅の方向に対して巡回を繰り返すことになる。

② 二分木と四則演算

㉔ 二分木で表現する算術式

二分木の葉の部分に変数を配置し、各節に四則演算の優先順位に従って、演算子を配置すると二分木を使用して算術式を表すことができる。

㉕ 四則演算式の表記法

四則演算式の表記法には次の3通りの方法がある。

㊦ 前置記法

演算子をオペランドの前に書く方法。 $+ a * - b c d$

㊧ 中置記法

演算子をオペランドの間に書く方法。 $(a + ((b - c) * d))$

㊨ 後置記法

演算子をオペランドの後に書く方法。 $a b c - d * +$

二分木を使用した算術式と二分木の巡回法を組み合わせると、㊨の表記法を求めることができる。

③ 逆ポーランド記法への活用

㉖ 四則演算と逆ポーランド記法

逆ポーランド記法は四則演算の後置記法である。四則演算式を逆ポーランド記法に変換する場合、二分木巡回の後行順を使用する。

変数を葉に、演算子を節に配置し、四則演算の優先順位が成立するように節間に枝を設ける。

㉗ $X = (A - B / C) * (D + E)$ の巡回木の作成要領

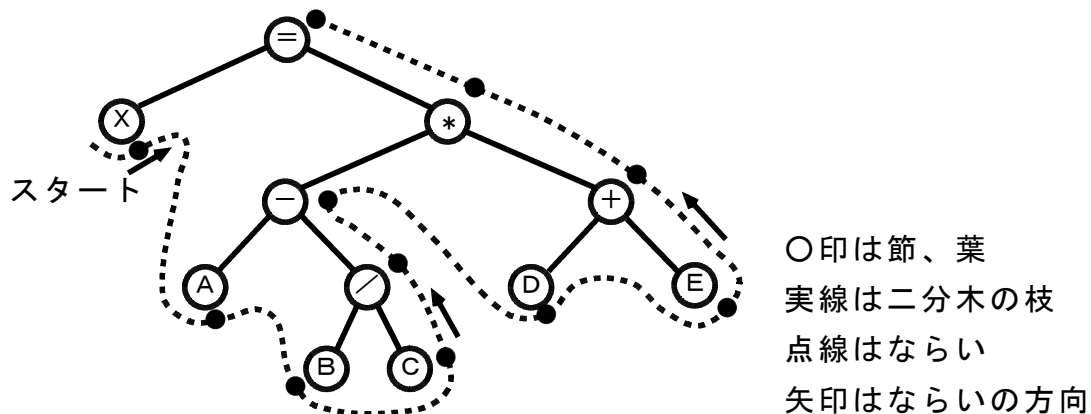
㊦ $=$ を根に配置する。

㊧ X を根の左部分木の葉に配置する。

- ㊸ 変数A、B、C、D、Eを右部分木の葉に配置する。
- ㊹ 四則演算子の優先レベルの高い葉から演算子で結合する。
- ㊺ 完成した二分木に後行順のならいを適用すると、 $XABC/-DE+*=$ の逆ポーランド記法を求めることができる。

㉔ $X = (A - B / C) * (D + E)$ の二分木のならいの実行順序

次の図のスタートから始め、点線の矢印に従って、=になるまで順次ノードを巡回する。



㉕ 二分木巡回法の後行順から逆ポーランド記法

二分木巡回法の後行順に従って、左から順次記号を並べると逆ポーランド記法を求めることができる。

$X = (A - B / C) * (D + E)$ の逆ポーランド記法は $XABC/-DE+*=$ となる。

㉖ スタックを利用した演算

逆ポーランド記法を左から順に解析し、数値であればスタックに格納し、演算子であれば、スタックから2つの数値を取り出して演算を行い、演算結果をスタックに格納する。2つの数値の演算は、(後から取り出した数値)演算子(先に取り出した数値)の方法で行う。

㉗ 演算のアルゴリズム

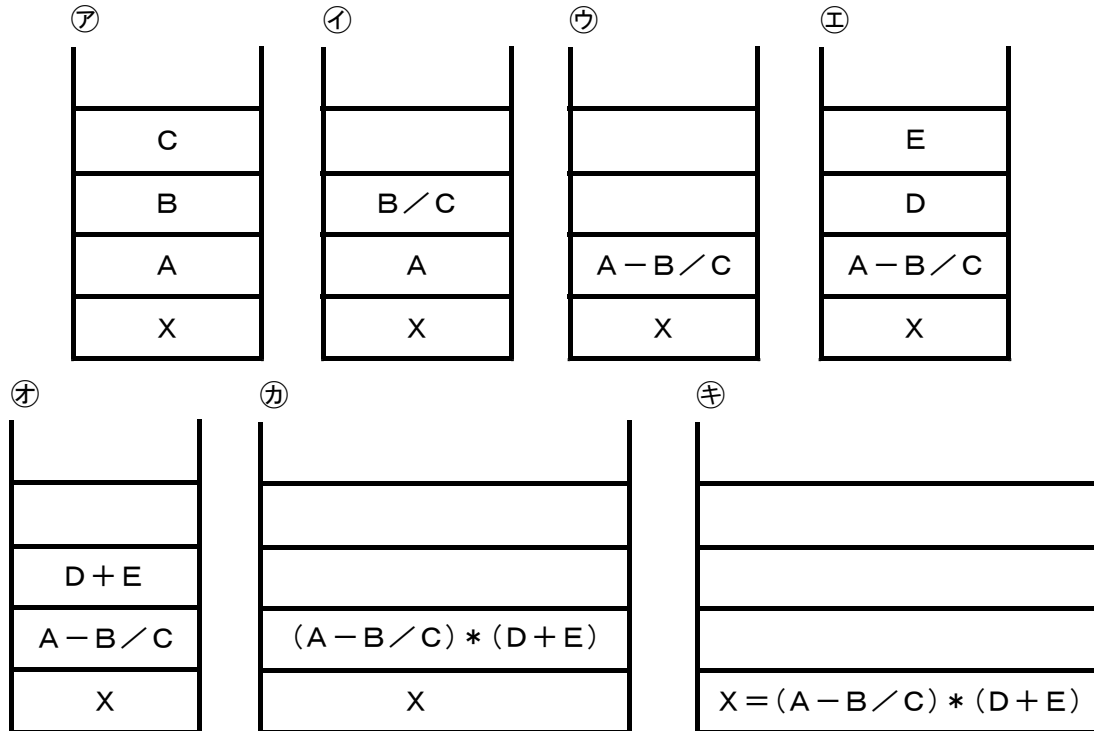
- ㊶ スタックポインタを初期化する。
- ㊷ 先頭から=が表れるまで、㊸、㊹の処理を繰り返す。
- ㊸ 数値ならば、スタックにPUSHする。
- ㊹ 演算記号ならば、次の処理を行う。
 - ❶ スタックから数値を2つPOPする。
 - ❷ 取り出した2数を(後から取り出した数値)演算子(先に取り出した数値)の方法で演算する。
 - ❸ 演算結果をスタックにpushする。
- ㊺ =の時演算結果をPOPして処理を終了する。

㉔ スタックを使用した演算の具体例

元の四則演算 $X = (A - B / C) * (D + E)$

逆ポーランド記法 $X A B C / - D E + * =$

スタックを使用した演算の状態の変化図と演算手順㉔～㉔を次に示す。



㉔ X、A、B、Cの順にスタックにPUSHする。

㉔ 次は演算子の/であるから、C、BをPOPし、 B / C を計算し、PUSHする。

㉔ 演算子-であるから、 B / C 、AをPOPし、 $A - B / C$ を計算し、PUSHする。

㉔ D、Eの順にスタックにPUSHする。

㉔ 演算子+であるから、E、DをPOPし、 $D + E$ を計算し、PUSHする。

㉔ 演算子*であるから、 $D + E$ 、 $A - B / C$ をPOPし、 $(A - B / C) * (D + E)$ を計算し、PUSHする。

㉔ 演算子=であるから、 $(A - B / C) * (D + E)$ 、XをPOPし、次の答えを求める。

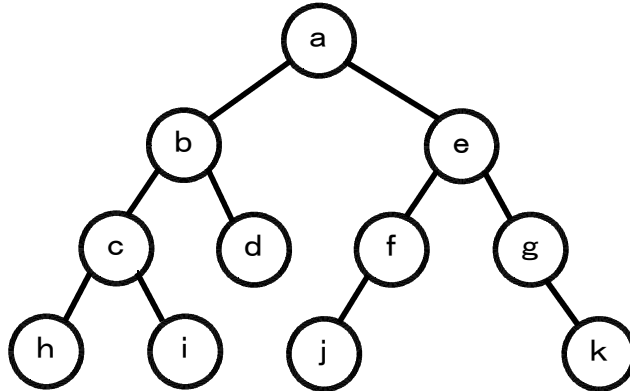
$$X = (A - B / C) * (D + E)$$

例題演習

それぞれの節から分岐する枝が2本以下である木を2分木という。2分木は一つの節とその左部分木と右部分木からなり、走査の方法にはその順序によって次の三つがある。

- (1) 前順：節点、左部分木、右部分木の順に走査する。
- (2) 間順：左部分木、節点、右部分木の順に走査する。
- (3) 後順：左部分木、右部分木、節点の順に走査する。

図に示す2分木に対して後順に走査を行い、節の値を出力した結果はどれか。



ア a b c h i d e f j g k
ウ h c i b d a j f e g k

イ a b e c h i d f j g k
エ h i c d b j f k g e a

解答解説

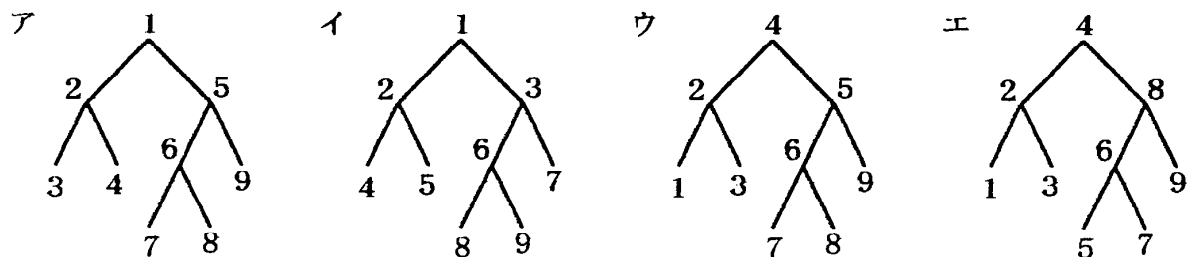
二分木の後行順操作に関する問題である。

後行順は、節点に来たときに自分を根とする部分木を巡回し終えてから、順位を割り当てる方法である。

文字の出力順序は、h i c d b j f k g e aとなり、求める答えはエとなる。

例題演習

2分木に対する系統的な巡回法のうち、幅優先順を表しているグラフはどれか。グラフ中の数字は巡回順序を示すものである。



解答解説

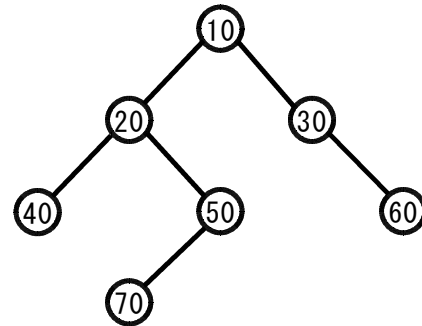
幅優先順のならいに関する問題である。

幅優先順は根から始まって、深さの浅い方からかつ左から巡回する方法である。木の幅の方向に対して巡回を繰り返すことになる。ルート部が1で次のレベルを左から右に向かって2、3となり、その次のレベルを左から4、5、6、7となり、その次のレベルが8、9となっている2分木となる。求める答えはイとなる。

例題演習

図の二分木において、ある走査を行ったところ、40、70、50、20、60、30、10の順に節の値が出力された。この走査法はどれか。

- ア 幅優先走査
- イ 中間順走査
- ウ 後行順走査
- エ 先行順走査



解答解説

二分木の巡回法に関する問題である。

深さ優先順は根から始めて左の子からかつ葉の方向に向かって、左の子から右の子の順に巡回を行う方法である。

巡回の順番を決める方法

先行順は、節点に来たときに先に順位を割り当てる方法である。

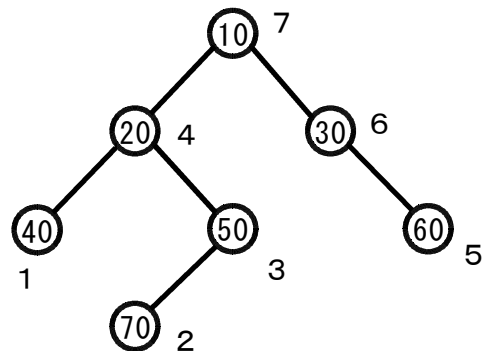
中間順は、節点に来たときにその左の子を根とする部分木を巡回し終えてから、戻ってきたときに順位を割り当てる方法である。

後行順は、節点に来たときに自分を根とする部分木を巡回し終えてから、順位を割り当てる方法である。

走査の順序を記入すると図の二分木のようなになる。

左右の部分木に番号が付加されると、その元の番号を付加する規則で各ノードの番号が付けられている。

この方式は後行順走査である。求める答えはウとなる。



例題演習

後置記法(逆ポーランド記法)では、例えば、式 $X = (A - B) \times C$ を $X A B - C \times =$ と表現する。次の式を後置記法で表現したものはどれか。

$$X = (A + B) \times (C - D \div E)$$

- | | |
|-----------------------------------|-----------------------------------|
| ア $X A B + C D E \div - \times =$ | イ $X A B + C - D E \div \times =$ |
| ウ $X A B + E D C \div - \times =$ | エ $X B A + C D - E \div \times =$ |

解答解説

逆ポーランド記法に関する問題である。

四則演算を逆ポーランド記法に変換するには二分木の後行順ならいを利用する。

二分木の葉の部分に変数をX A B C D Eの順に並べる。四則演算の演算子の優先度レベルの順位の高いものから順次、次のように演算子で結合する。

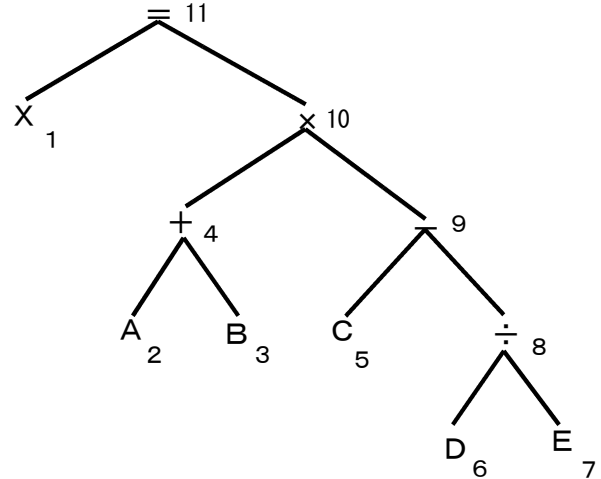
- ① A Bを+で結合する。
- ② D Eを÷で結合する。
- ③ Cと÷を一で結合する。
- ④ +と－を×で結合する。
- ⑤ Xと×を＝で結合する。

完成した二分木に後行順ならいを実行すると、図の二分木に示すノードの順番になる。

このノードの順番に変数、演算子を並べると答えが次のように求まる。

$$X A B + C D E \div - \times =$$

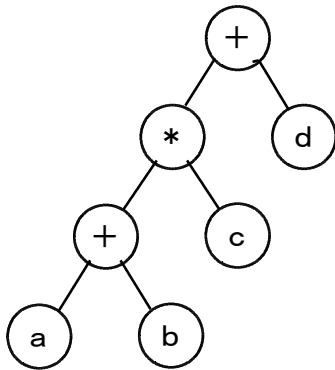
となり、求める答えはAとなる。



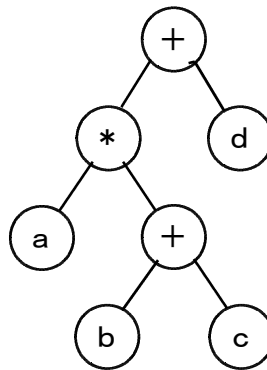
例題演習

$a * (b + c) + d$ を木構造で表現したものはどれか。

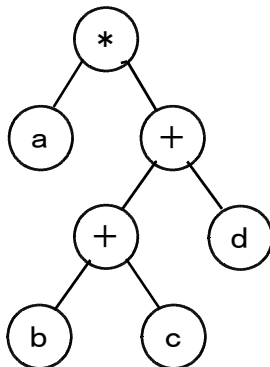
ア



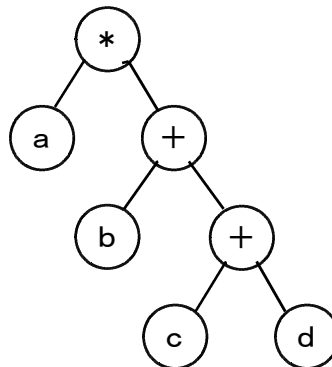
イ



ウ



エ



解答解説

二分木を利用した算術式の表現法に関する問題である。

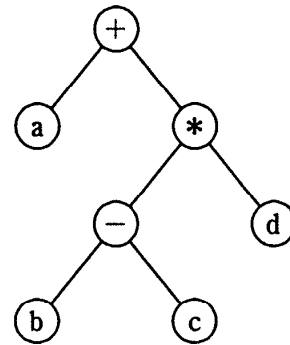
演算レベルが最も高いのは、 $(b + c)$ 、その次が、 $a * (b + c)$ の $a *$ になる。

最後が、 $+ d$ となる。この順序で二分木の下位のレベルから割り当てられているのは、イである。求める答えはイとなる。

例題演習

2 分木の各ノードがもつ記号を出力する再帰的なプログラム Proc(ノード n) は、次のように定義される。このプログラムを、図の 2 分木の根に適用したときの出力はどれか。

```
Proc(ノード n) {  
    n に左の子 l があれば Proc(l) を呼び出す  
    n に右の子 r があれば Proc(r) を呼び出す  
    n に書かれた記号を出力する  
}
```



- ア $b - c * d + a$
- イ $+ a * - b c d$
- ウ $a + b - c * d$
- エ $a b c - d * +$

解答解説

2 分木に関する問題である。

プログラム Proc() を適用すると、次の順序で処理され記号を出力した時、元の呼出点に戻る。

- ① +の左の子 a を呼出、出力、+に戻る。
- ② +右の子 * 呼出
- ③ *の左の子 - を呼出
- ④ -の左の子 b、を呼出、出力、-に戻る。
- ⑤ -の右の子 c、を呼出、出力、-に戻る。
- ⑥ -を出力、*に戻る。
- ⑦ *の右の子 d、を呼出、出力、*に戻る。
- ⑧ *を出力
- ⑨ +を出力し終了する。

出力は a b c - d * + の順になる。求める答えはエとなる。

例題演習

逆ポーランド表記法(後置表記法)で、 $E F - G \div C D - A B + \div +$ と表現される式はどれか。

- ア $((A + B) + (C - D)) \div G - (E \div F)$
- イ $((A + B) \div (C - D)) + G \div (E - F)$
- ウ $((E - F) \div G) + ((C - D) \div (A + B))$
- エ $((E - F) \div G) \div ((C - D) + (A + B))$

解答解説

逆ポーランド記法(後置表記法)に関する問題である。

逆ポーランド記法を利用して、左から順次演算子の前の2つの変数で式を求める。

- ① $E F -$ から $E - F$
- ② $(E - F) G \div$ から $(E - F) \div G$
- ③ $C D -$ から $C - D$
- ④ $A B +$ から $A + B$
- ⑤ $(C - D)(A + B) \div$ から $(C - D) \div (A + B)$
- ⑥ ②、⑥の結果を利用して、 $((E - F) \div G)((C - D) \div (A + B)) +$ から次の式が求まる。
 $((E - F) \div G) + ((C - D) \div (A + B))$

求める答えはウとなる。

例題演習

$A = 1, B = 3, C = 5, D = 4, E = 2$ のとき、逆ポーランド表記法で表現された式 $A B + C D E / - *$ の演算結果はどれか。

ア -12

イ 2

ウ 12

エ 14

解答解説

逆ポーランド記法に関する問題である。

$A B + C D E / - *$ に値を代入すると、 $1 3 + 5 4 2 / - *$ となる。

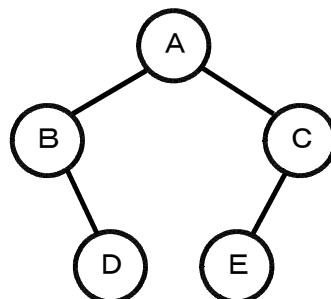
演算は次のようになる。

$$(1 + 3) \times (5 - 4 / 2) = 4 \times (5 - 2) = 4 \times 3 = 12$$

求める答えは12となる。

例題演習

2分木の節に付けられた記号を印字する。印字の順序は各節について、その節の左部分木、その節、その節の右部分木の順である。図の2分木について印字した結果はどれか。



ア $A B D C E$

ウ $B D A E C$

イ $B D A C E$

エ $D B A C E$

解答解説

二分木に関する問題である。

印字の順序は、

その節の左部分木、その節、その節の右部分木
であるから、B D A E Cとなり、求める答えはウとなる。

例題演習

次に示す計算式と逆ポーランド表記法の組合せのうち、適切なものはどれか。

	計算式	逆ポーランド表記法
ア	$((a+b)*c)-d$	$abc * + d -$
イ	$(a+(b*c))-d$	$ab + c * d -$
ウ	$(a+b)*(c-d)$	$abc * d - +$
エ	$a+(b*(c-d))$	$abcd - * +$

解答解説

四則演算式から逆ポーランド表記法を作成する問題である。

逆ポーランド記法を作成する場合、変数、演算子を節とする二分木を作成し、作成した二分木の各ノードに後行順に従って節番を付け番号順に変数、演算子を並べる方法がある。

四則演算の計算式から直接、逆ポーランド記法を作成する手順を検討する。

- ① 計算式の変数のみを先頭から順に並べる。
- ② 四則演算の優先度の高い演算に着目する。
- ③ ②で着目した演算の演算子を関係する2つの変数の次に挿入する。
- ④ 次の優先度の高い演算に着目し、②、③の処理を行う。
- ⑤ ④の処理において、既に演算した結果を利用する場合、挿入した演算子の後にある関係する変数の後に新しい演算子を挿入する。
- ⑥ 既に演算した結果のみを利用したり、挿入した演算子よりも関係する変数が前にある場合は、一つ前に挿入した演算子の後に新しい演算子を挿入する。
- ⑦ ②～⑤の処理を計算式の最後まで繰り返す。

ア～エの計算式に、①～⑦の処理を実行すると次のようになる。

アは $a b + c * d -$ 、イは $a b c * + d -$ 、ウは $a b + c d - *$ 、エは $a b c d - * +$ となる。
求める答えはエとなる。

① バランス木と完全二分木

① バランス木

二分木や多分木はデータの挿入や削除を繰り返していると、木の形が変形しアンバランスな木の形になる。このようなアンバランスな状態を避ける仕組みを持った木がバランス木である。バランス木はデータの挿入や削除によって、一定の条件以上になると、分裂や統合によって木全体の構造が再調整する機能をもっている。バランス木は探索木的一种である。

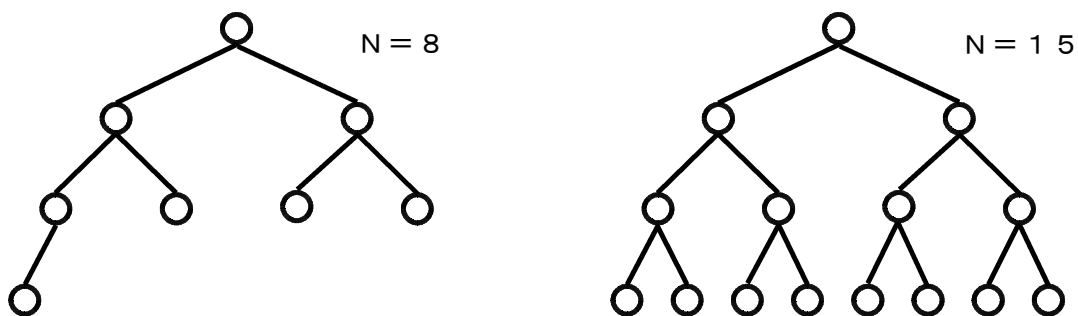
② AVL木

AVL木は二分木をもとにしたバランス木である。各節点において、その部分木の左と右の深さの差が高々一つしか異ならないような二分木である。各部分木は左と右の深さの情報をもっている。データの挿入や削除が行われる際に、木の構造を調整する。

③ 完全二分木

根から葉までの経路の長さが等しい二分探索木を完全二分木という。完全二分木が成立するのは、節の数が $2^n - 1$ 個の時である。通常は、根から最も遠い葉までの経路の長さと、根から最も近い葉までの経路の長さの差が1以下である二分木を完全二分木と定義する。

④ 階層数 $K = 4$ の場合の完全二分木



⑤ 完全二分木の階層と節点の個数

階層数	節点の個数		数式
	最小	最大	
1	1	1	$1 = 2^1 - 1$
2	2	3	$3 = 2^2 - 1$
3	4	7	$7 = 2^3 - 1$
4	8	15	$15 = 2^4 - 1$
$\vdots$	$\vdots$	$\vdots$	

$$\begin{array}{lll}
 K-1 & 2^{K-2} & 2^{K-1}-1 \\
 K & 2^{K-1} & 2^K-1
 \end{array}$$

階層数を K とし、節点の個数を N とすると、次の式が成立する。

$$N = 2^K - 1$$

② 多分木

① m分木

m分木は一つの節が最大m個の子をもつことができる木構造である。

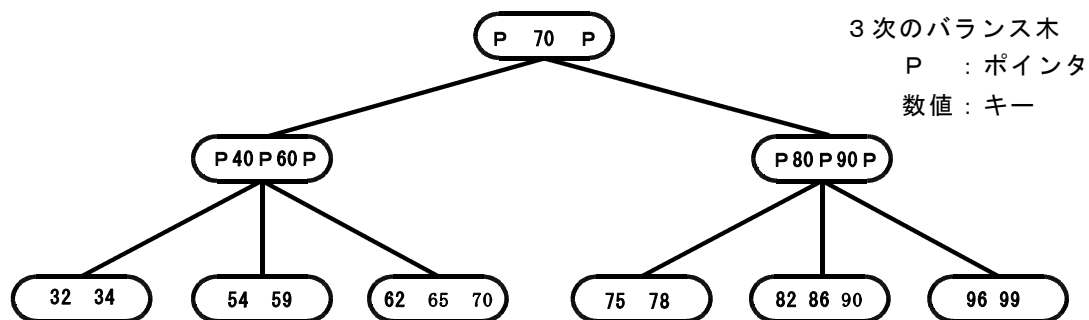
② m次のバランス木の条件

- ㊦ 根は2～m個の子を持つ。
- ㊧ 根と葉以外の節は $(\lceil m/2 \rceil \sim m)$ 個の子を持つ。 $\lceil m/2 \rceil$ は $m/2$ 以上の最小の整数。
- ㊨ 根からすべての葉までの経路は長さは等しい。
- ㊩ k個の子を持つ親は、 $(k-1)$ 個のキーを持っている。
- ㊪ すべての葉は同じレベルにあって、キーは持たない。

③ バランス木の構造

2分木は節が値を持っているがバランス木は値を持つのは葉だけであり、葉以外の節は値を持たずキーだけを持っている。キーとはポインタで示す節の持つ値の範囲を示す。

④ レベル $k=2$ の場合のバランス木



③ 5次のバランス木

① 5次の多分木の構造

5次の多分木は各節にキーが4個、ポインタが5個あるバランス木である。すべてのレコー

ドは葉の部分にあって、各ノードに配置されたキーを辿ることによって目的のレコードに到着することができる。

⑥ 深さが2の場合のキーの数

ルート部のキーの数は4個である。次のレベルはノードが5個になる。

更に、その下のレベルのノードの数は、 $5 \times 5 = 25$ となり、ルート部からのノードの総数は $1 + 5 + 25 = 31$ 個となる。

各ノード部にはキーが4個格納されるから、格納できるキーの総数は

$$(1 + 5 + 25) \times 4 = 31 \times 4 = 124 \text{個}$$

となる。

⑦ 深さが3の場合のキーの数

ノードの数が $25 \times 5 = 125$ 個追加されて、ノードの総数は、 $31 + 125 = 156$ となり、格納できるキーの総数は

$$(1 + 5 + 25 + 125) \times 4 = 624 \text{個}$$

となる。

レベルが3の場合、葉は125個で、それぞれの葉の部分に5個のレコードが含まれ、最右端の葉のみが4個のレコードとなる。従って、全レコード数は

$$124 \times 5 + 4 = 620 + 4 = 624$$

となり、レコードの総数とキーの総数が一致する。

④ ヒープ木の定義と操作

① ヒープ木の定義

ヒープ木は完全二分木の各節に一連番号を付け、その番号をもとに、完全二分木のデータ構造を配列で表現したものである。二分木に番号をつけ、どの親子の値をとっていても必ず「親の値 < 子の値」(または「親の値 > 子の値」)の関係になっている配列をヒープ木という。親の節の値が子の節の値より小さくなるようにすれば、根の値は二分木の中で最小になる。逆にすれば最大になる。

② 一連番号付けの規則

㊦ 根には番号1をつける。

㊧ 番号kの節の左の子には2k、右の子には2k+1という番号をつける。

㊨ 親の節の値は子の節の値よりも小さい(または大きい)値となるようにする。

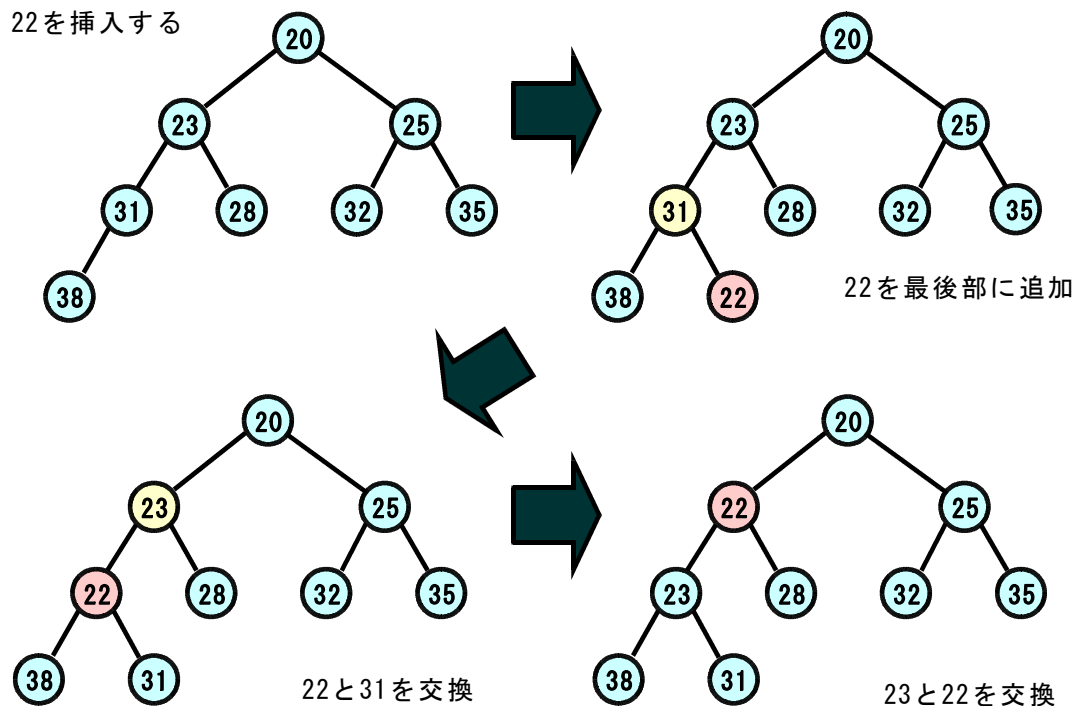
㉔ ヒープの探索と挿入

ヒープの探索は線形探索と同じである。節につけられた一連の番号を順次たどっていきながら、探索データと各要素を比較し値が一致すれば探索成功、最後まで探索して一致するデータが存在しなければ探索不成功になる。ヒープ木に値22のデータを挿入する場合の具体例を次に示す。挿入の手順は矢印の方向に進められる。

㉕ 親の値 $\leq$ 子の値の場合の挿入手順

- ㊦ 挿入する要素を最後部へ無条件に追加する。
- ㊧ 追加した要素の値がその親の値よりも小さいと親と子を交換する。
- ㊨ 新しく親となった値とその親の値を比較して、子の方が小さければその親と子を交換する。
- ㊩ この比較・交換を根に向かって行い、「親の値 $\leq$ 子の値」の関係になるまで続ける。

値22を挿入する手順を次に示す。



㉖ ヒープの削除

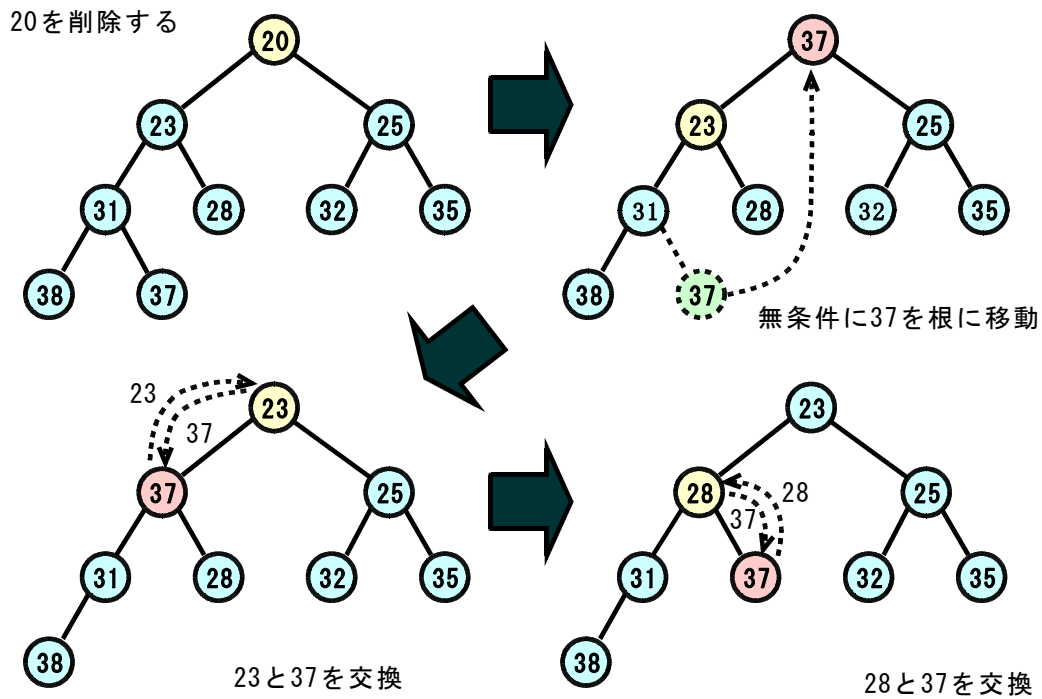
ヒープの削除は最小の値(または最大の値)の要素を取り出すことである。最小の値の要素(最大の値の要素)はヒープの性質上、根にある。従って、ヒープの削除は根の削除に相当する。

㉗ 親の値 $\leq$ 子の値の場合の削除手順

- ㊦ 根の部分の値を削除する。
- ㊧ 根の部分にヒープの最後の要素を移動する。

- ㊦ 根から順次「親の値 $\leq$ 子の値」が保たれるように要素の値を比較・交換する。
- ㊧ ヒープ木のすべての親子関係が「親の値 $\leq$ 子の値」になると、その木が新しいヒープ木となる。

値20を削除する手順を次に示す。



例題演習

5次のB木（各節点から出る枝の数 ≤ 5 、または各節点に格納されているキーの数 ≤ 4 ）で、レベル2（深さ2でも同じ）までに格納できるキーの数はどれか。

- ア 96 イ 100 ウ 124 エ 125

解答解説

5次のB木の深さのレベル2の最大キー数を求める問題である。

5次のB木の特徴

- ① メモリの使用効率を上げるため、各節点のポインタの数は、3個以上5個以下である。
- ② 根を除く要素のキーの数は4個。従って、2個以上4個以下である。
- ③ 根のポインタ数は2個以上、要素のキーの数は1個以上
- ④ データを追加・削除するたびに、節点の要素数は5個以上、または2個未満になると自動的に木の構造を変更し、葉のレベルが同一になるように調整する。

根の部分に4個、1レベルの部分は節から枝が5個出るから、 $5 \times 4 = 20$ 、2レベルの部分は更に各節から5個の枝が出るから、 $5 \times 6 \times 4 = 100$ 、従って、キーの全部の数は

$$100 + 20 + 4 = 124 \text{個}$$

となる。求める答えはウとなる。

例題演習

どの節から見ても、左右の部分木の高さの差が高々1しかない2分木(平衡2分木)において、節が七つの時、この2分木の高さの最低は2になるが、最高はいくつになるか。

- ア 2
- イ 3
- ウ 4
- エ 5

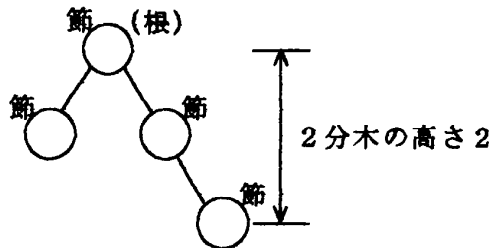
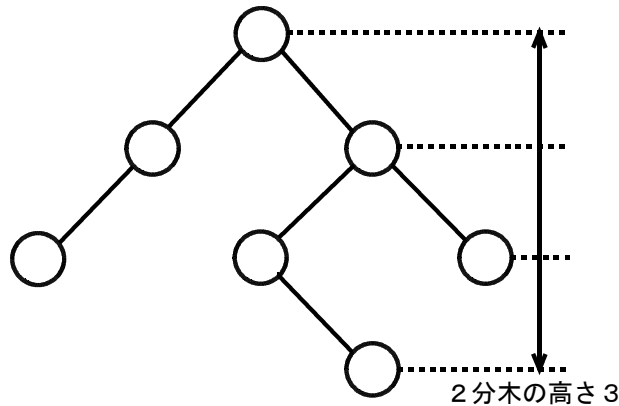


図 平衡2分木の例

解答解説

平衡2分木に関する問題である。



どの節から見ても左右の部分木の差が高々1しか差がないという平衡2分木の定義からすると、節の数が計7個の場合の平衡2分木の例は図のようになる。図から、2分木の高さの差は3となり、求める答えはイである。

例題演習

B木に関する記述のうち、正しいものはどれか。

- ア B木内の各要素は、追加された順に格納されている。
- イ 根からそれぞれの葉までのレベル(深さ)は、要素のキー値に偏りがあると、一定にならない。
- ウ 要素の削除に伴って、部分木の要素数が一定値を下回るときは、隣の葉や節を含めて再構成される。
- エ 要素は、必ず新しい葉に対して追加される。

解答解説

バランス木に関する問題である。

2分木や多分木は要素の挿入・削除を繰り返していくと、木の形が変形しアンバランスな木の形になる。このようなアンバランスな状態を避けるのがバランス木である。

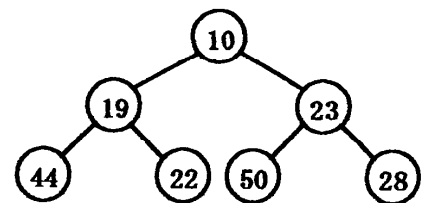
バランス木は探索の回数を平均化するため、データの追加、削除によりバランスが崩れると、分裂・統合により木のバランスを保つように木全体の構造を再調整する機能を持っている。即ち、要素の削除に伴って、部分木の要素数が一定値を下回ると、隣の葉や節を含めて再調整する。

求める答えはウとなる。

例題演習

次の図の2分木はヒープである。これを配列で表したものはどれか。

ア	10	19	23	44	22	50	28
イ	10	19	22	23	28	44	50
ウ	44	19	22	10	23	50	28
エ	44	22	50	28	19	23	10



解答解説

ヒープ木に関する問題である。

ヒープ木は根の部分に添字の1として、各レベルの左から右に向かって添字の番号を付ける。下位のレベルの値は上位のレベルの値より大きい(または小さい)の特徴がある。

ヒープ木の節の値は次の順に並べられる。

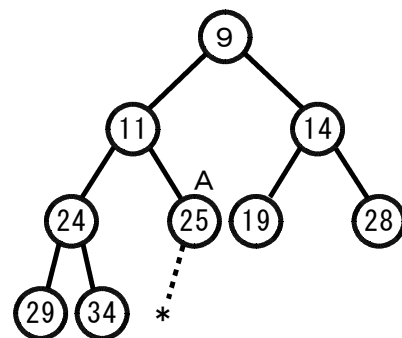
10→19→23→44→22→50→28

求める答えはアである。

例題演習

親の節の値が子の節の値より小さいヒープがある。このヒープへの挿入は、要素を最後部に追加し、その要素が親よりも小さいとき親と子を交換することを繰り返せばよい。次のヒープの*の位置に7を追加したとき、Aの位置にくる要素はどれか。

- ア 7
- イ 9
- ウ 11
- エ 25



解答解説

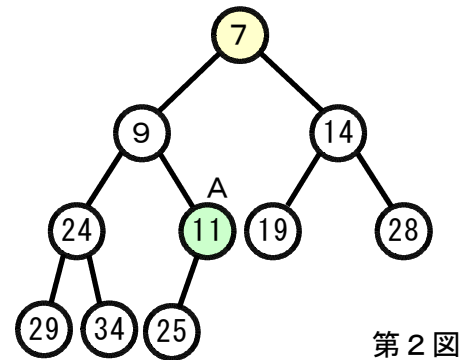
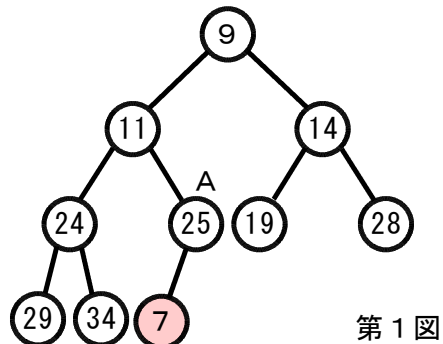
ヒープ木の挿入に関する問題である。

*の位置に7を追加した図は、第1図の2分木になる。

ヒープの規則を用いて、ノードの値を交換すると次のようになる。

- ① 7と25を交換する。
- ② 7と11を交換する。
- ③ 7と9を交換する

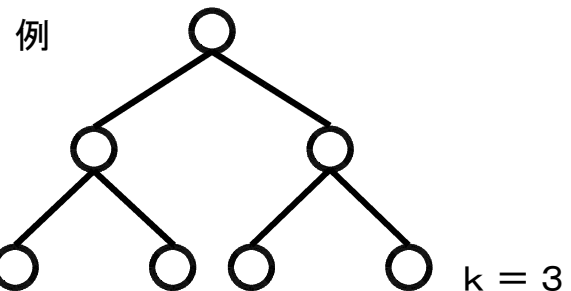
交換後のヒープ木は第2図のようになる。Aのノードにくる値は11となる。求める答えはウである。



例題演習

すべての葉をもつ完全2分木がある。この完全2分木で成り立つ関係式はどれか。ここで、 n は節点の個数、 k ($k \geq 1$)は根から葉までの階層数を表す。例の階層数(k)は3である。

- ア $n = k(k-1) + 1$
 イ $n = k(k-2) + 3$
 ウ $n = 2^k - 1$
 エ $n = 2^k + 1$



解答解説

バランス木の節点に関する問題である。

階層と節点の個数の関係

階層数	節点の個数		数式
	最小	最大	
1	1	1	$1 = 2^1 - 1$
2	2	3	$3 = 2^2 - 1$
3	4	7	$7 = 2^3 - 1$
4	8	15	$15 = 2^4 - 1$
$\vdots$	$\vdots$	$\vdots$	
$K-1$	2^{K-2}	$2^{K-1} - 1$	
K	2^{K-1}	$2^K - 1$	

階層数を K とし、節点の個数を N とすると、次の式が成立する。

$$N = 2^K - 1$$

求める答えはウとなる。

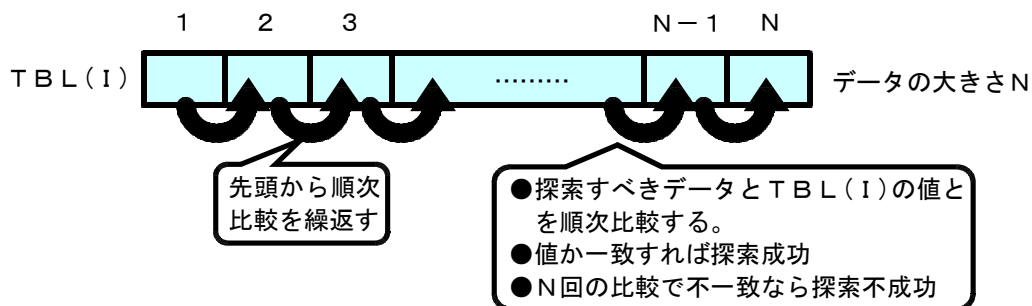
txt01022 アルゴリズム

txt010221 探索アルゴリズム

① 線形探索アルゴリズム

① 線形探索法

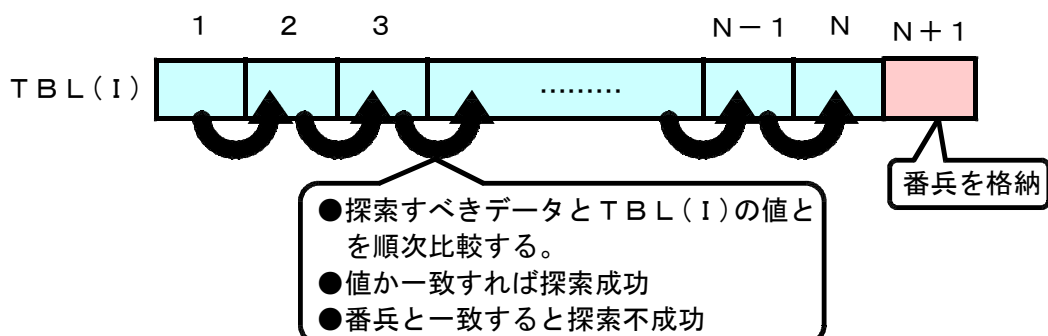
線形探索は配列の先頭から（または後方から）探索データを探し出すまで、または、配列の最後に（先頭に）至り探索データがないことが判るまで、配列の要素を1つずつ調べる方法である。探索すべきデータと配列の各要素の値の比較を繰り返す操作になる。先頭の位置または後方の位置が既知で、データの個数が決まっている場合に可能な操作である。



② 線形探索の手順

- ㊦ 初期条件の設定： $1 \rightarrow I$ （または $N \rightarrow I$ ）
- ㊧ 添字の範囲のチェック： $I > N$ （または $I > 0$ ）
- ㊨ データの比較：TBL(I) : K (探索データ)
- ㊩ 添字のインクリメント： $I + 1 \rightarrow I$ （または $I - 1 \rightarrow I$ ）
- ㊪ 終了条件の判定：TBL(I) = K または $I > N$ ($I < 0$)

③ 番兵を使用した線形探索

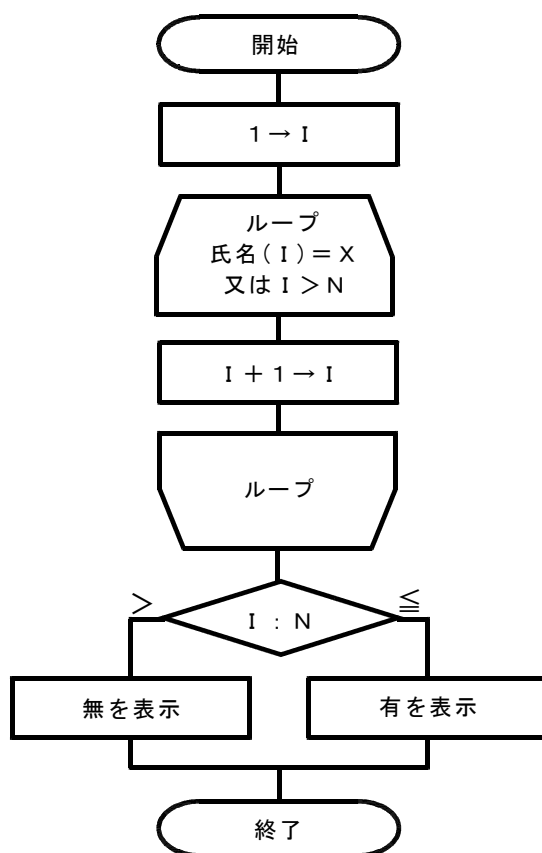


線形探索には番兵という考え方がある。通常の線形探索は対象のデータの個数が既知の場合に可能であるが、データの個数が未知の場合には探索が終了しない問題がある。そこで、終了条件として番兵を利用する。探索するデータを番兵として最後のデータの次に置く。番兵の位置まで探索データと配列の要素の値の比較が進むとデータが必ず一致し探索を終了することが可能になる。番兵を使用することによって、通常探索の場合の添字の範囲のチェックが不要となり、操作のための比較回数が減少する。

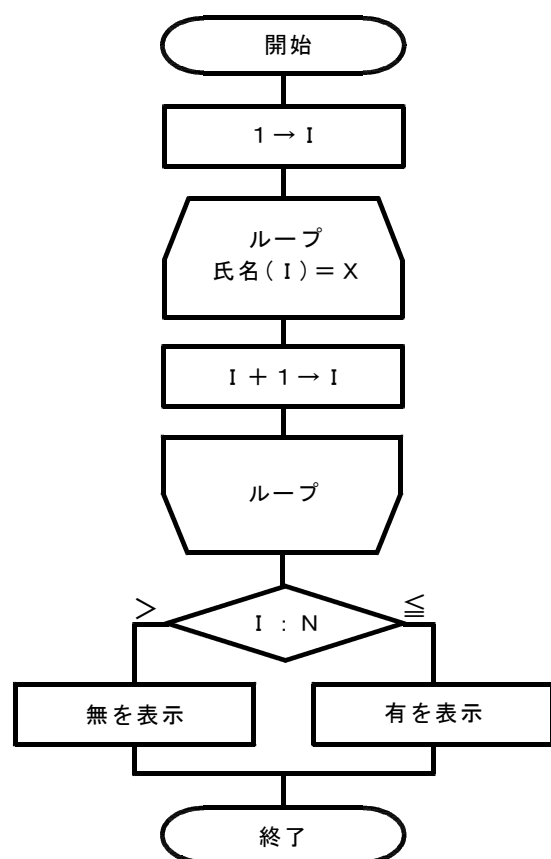
④ 線形探索の流れ図

両者の流れ図はループからの脱出条件が異なる。

通常の線形探索



番兵を使用した線形探索



⑤ 線形探索の比較回数

データの個数がNの場合、平均比較回数は $(N + 1) / 2$ 、最大比較回数はN回となる。Nが十分に大きい場合、平均比較回数は $N / 2$ となる。

⑥ 線形探索と番兵の比較回数

配列に10名の氏名が格納されている場合の通常の線形探索と番兵を使用した線形探索の最悪の場合の比較回数を求める。番兵を使用した線形探索の場合、探索データが配列の中に必ずあり、配列の領域 $I > 10$ の判定が不要となるため、比較回数が減少する。

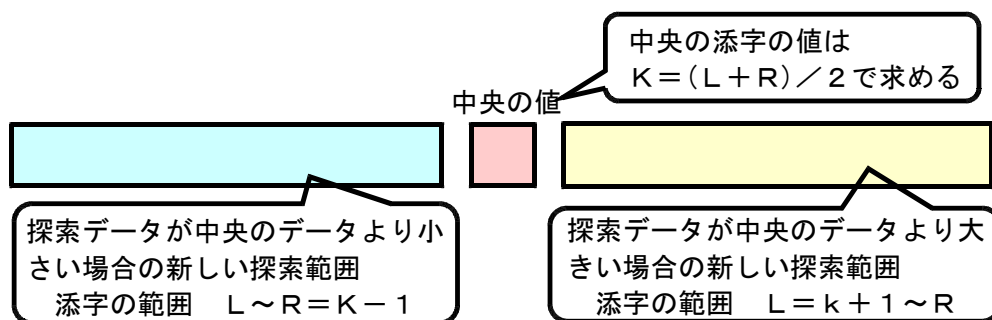
ステップの内容	通常の場合	番兵の場合
添字 I の初期設定	1	1
配列の領域 $I > 10$ の判定	11	0
氏名 (I) と探索データ X の比較	10	11
$I + 1 \rightarrow I$ の演算	10	10
$I > 10$ の判定	1	1
結果の表示	1	1
全比較回数	34	24

② 二分探索アルゴリズム

① 二分探索法

配列が昇順または降順に整列されている場合、比較対象のデータを配列の中央の添字の値に設定し探索データと比較する。一致しない場合、整列済み配列を2分割して、探索データがどちらの系列に属するかを調べるため中央の値と比較する。昇順に整列されている場合、探索データが中央の値よりも小さい場合には、中央の値よりも前の範囲、そうでない場合には後ろの範囲を新しい探索範囲に設定する。再び、2分割して属する範囲を調べる。

二分探索法は、この操作を繰り返してデータを探索する方法である。探索の対象となる配列が、あらかじめ昇順(降順)に並んでいるときに用いることができる。



② 二分探索法の手順

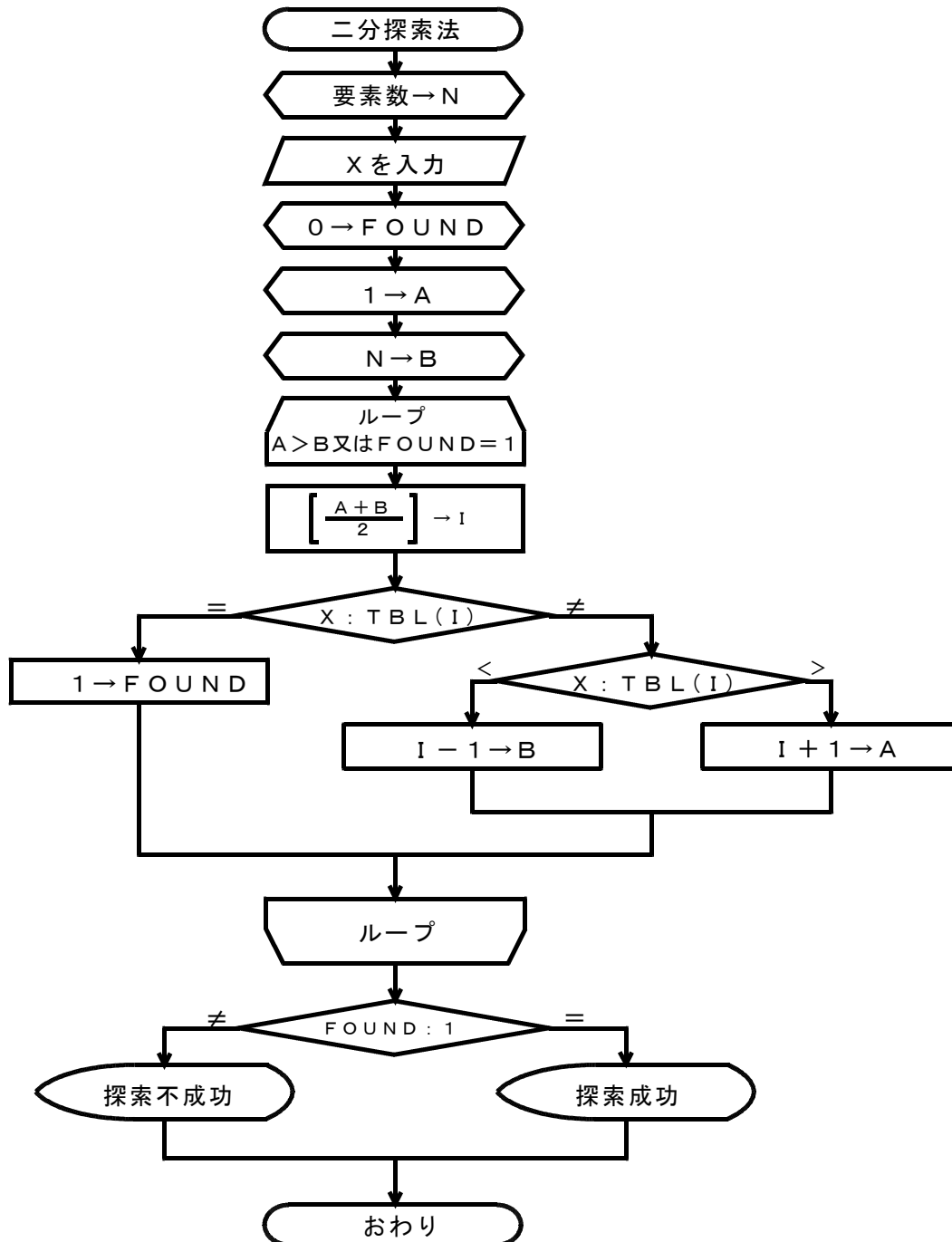
- ㊦ 対象の配列を昇順または降順に並べる。
- ㊧ 探索する範囲、探索下限の添字A、探索上限の添字Bを設定する。
- ㊨ 比較対象の中央の要素の添字 $K = [(A + B) / 2]$ を決める。
- ㊩ Kの要素の値と探索する値が一致すると探索成功。探索を終了する。
- ㊪ Kの要素の値 > 探索する値のときは、探索上限の添字 $B = K - 1$ を求め、 $A \sim B$ を新しい探索範囲とし、処理㊨に移る。
- ㊫ Kの要素の値 < 探索する値のときは、探索下限の添字 $A = K + 1$ を求め、 $A \sim B$ を新しい

探索範囲とし、処理⑤に移る。

⑤ 探索下限の添字Aと探索上限の添字Bが、 $A \leq B$ の間、⑥～⑦を繰り返す。

⑦ 探索下限の添字Aと探索上限の添字Bが、 $A > B$ となると探索不成功。対象の配列の中に探索する値が存在しないので探索を終了する。

㉔ 二分探索の流れ図



㉕ 二分探索の終了条件

探索する範囲の探索下限の添字をA、探索上限の添字をBとすると、 $A > B$ になると、探索

を終了する。 $A > B$ であって、 $A \geq B$ でないことに注意する必要がある。

⑤ データ個数がNの場合の比較回数

平均比較回数は $\lceil \log_2 N \rceil$ 回、最大比較回数 $\lceil \log_2 N \rceil + 1$ 回となる。

平均比較回数は探索するデータが見つかる場合の最大の比較回数であり、最大比較回数は探索するデータが見つからない場合の比較回数である。最大比較回数は、平均比較回数 + 1 となる。

③ ハッシュ法

① ハッシュ法の定義

配列、リスト、二分木のデータ構造は一定の順序で処理する必要があったが、ハッシュ法はデータの値から直接、格納位置を計算することができ、データ参照を高速に行う。キー値 X を関数 $H(X)$ を用いて、配列の添字やアドレスに変換する。関数 $H(X)$ をハッシュ関数といい、ハッシュ関数が返す値をハッシュ値という。ハッシュ関数を用いて、データを格納する添字やアドレスを求めることをハッシングという。ハッシュ法の計算量は通常、 $O(1)$ である。

② 衝突(コリジョン)

異なったキー値から同じハッシュ値が求まることを衝突という。先に格納されていたデータをホームデータ、後のデータをシノニムデータという。衝突が発生した場合、この衝突を回避する方法に、チェーン法とオープンアドレス法がある。

③ チェーン法

データを格納する場合、同じハッシュ値を持つデータをポインタを利用してリストでつなぐ方法である。

④ チェーン法のデータ格納手順

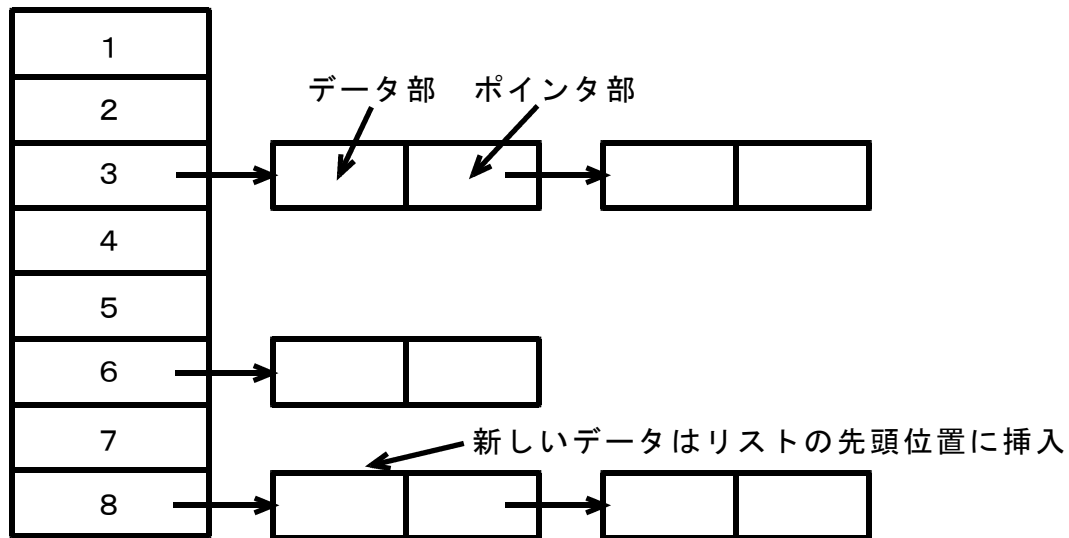
- ㊦ ハッシュ関数を用いてデータを格納する位置を計算する。
- ㊧ データの格納箇所が空であれば、データをそのアドレスにセットし、ハッシュ表のポインタ部にアドレスを設定する。
- ㊨ 空でなければ、リストの先頭に新しいデータを挿入する要領でデータを挿入する。

⑤ チェーン法のデータ探索手順

- ㊦ データとハッシュ関数からハッシュ値を求め、格納位置を知る。
- ㊧ ハッシュ表のアドレスから最初のデータを探し、探索するデータと比較する。
- ㊨ 一致すれば探索成功。

- ⑤ 不一致ならば、ポインタ部から次の比較データの格納位置を求め、探索データと比較する。
データが見つかるか、データがなくなるまで同じ操作を繰り返す。

ハッシュ表



⑥ チェーン法探索の計算量

格納位置を決める計算量は $O(1)$ で、チェーンでつながっているデータを探索する線形探索は $O(N/B)$ となり、探索全体では $O(1 + N/B)$ となる。 N はデータの数、 B は格納箇所数である。

⑦ チェーン法挿入の計算量

リストの先頭に格納する計算量は $O(1)$ である。重複チェックを行う場合は全体で $O(1 + N/B)$ となる。

⑧ オープンアドレス法

オープンアドレス法でデータを格納する場合、衝突が発生すると再ハッシュで対処し、データをハッシュ表の中に全て格納する方法である。再ハッシュの方法は、「元のハッシュ値 + 1」を用いる。衝突すると、再ハッシュ値を求め、そのデータ部にデータが入っていると、「元のハッシュ値 + 1」で次の格納位置を求める。空いている格納位置が見つかるまで再ハッシュを繰り返し位置が決まれば挿入する。

探索の場合も、ハッシュ値を求め格納位置を求めると、その位置のデータの値と比較して同じであれば、探索成功となる。異なれば、「元のハッシュ値 + 1」で再ハッシュして、次の格納位置のデータの値と比較する。データの値が一致するか、空のデータ部が現れると探索終了になる。

④ アルゴリズムと実行効率

④a アルゴリズムと計算量

同一のデータ構造を使用してもアルゴリズムが異なると実行効率が異なる。実行効率は命令実行数や比較回数で判断することができる。アルゴリズム作成時には、実行効率を高めるために計算量が小さくなるように論理を組み立てる。

④b アルゴリズム作成時の留意点

アルゴリズム作成時には次の留意点を検討し、実行効率の向上を図る必要がある。

- ㊦ 正しい理論に従って手順を組み立てる。
- ㊧ わかりやすく修正しやすい手続きによって組み立てる。
- ㊨ 実行効率を最適にする。
- ㊩ アルゴリズムを改善する。
- ㊪ 多数の条件の整理や、条件判断の順序を最適にする。

⑤ 探索法と最大比較回数

⑤a 線形探索の最大比較回数の求め方

データ数 n の場合、線形探索の最大比較回数は n 回である。

⑤b 二分探索の最大比較回数の求め方

データ数 n の場合、二分探索の平均比較回数を K とすると、次の式が成り立つ。

$$2^K \leq n < 2^{K+1}$$

両辺の 2 を底とする対数をとると

$$\log_2 2^K \leq \log_2 n < \log_2 2^{K+1}$$

$$K \leq \log_2 n < K + 1$$

$$K = \lceil \log_2 n \rceil$$

平均比較回数は $\lceil \log_2 n \rceil$ となる。

平均比較回数よりも、更に、1 回 2 分割の回数を増やすと、データ数が n を超えるため、その時の比較回数が最大比較回数となり、 $K + 1 = \lceil \log_2 n \rceil + 1$ となる。

$n = 10^8$ の時は

$$\begin{aligned} \lceil \log_2 10^8 \rceil + 1 &= \lceil 8 / \log 2 \rceil + 1 = \lceil 8 / 0.301 \rceil + 1 = \lceil 26.6 \rceil + 1 \\ &= 26 + 1 = 27 \end{aligned}$$

最大比較回数は 27 回となる。

同様にして、 $n = 1000$ 、 $n = 100000$ の場合について計算すると、次に示す表を得ることができる。

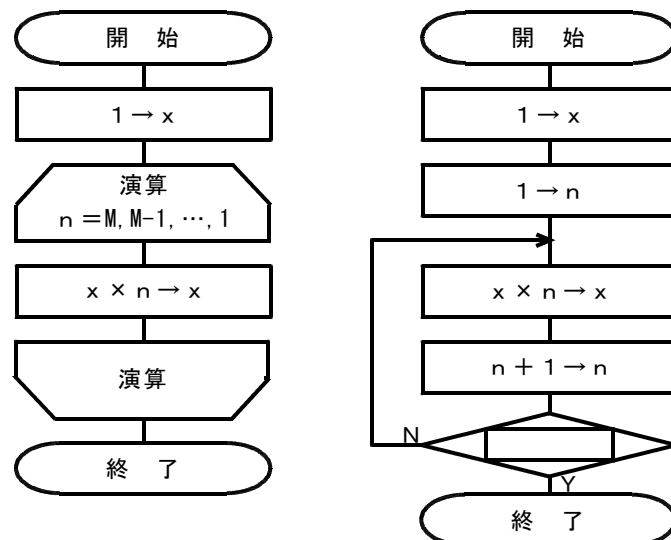
㉔ 線形探索、二分探索、ハッシュ法の最大比較回数

データ数	最大比較回数		
	線形探索	二分探索	ハッシュ法
1000	1000	10	1
10000	10000	14	1
10^8	10^8	27	1
N	N	$\log_2 N + 1$	1

データ数が大きくなると、2分探索の最大比較数は、線形探索の最大比較数に比べて大幅に少なくなる。2分探索の場合、データ数が1000程度の場合、データ数が10倍になっても、最大比較回数は1.4倍程度の増加である。データ数が1万から1万倍の1億に増加しても2分探索の場合、最大探索回数は2倍程度に増加するのみである。

例題演習

次の二つの流れ図に示したアルゴリズムを実行したとき、結果のxの値が同じになるようにしたい。判断記号の□に入れる条件として正しいものはどれか。



ア $n > M$

ウ $n > M - 1$

イ $n > M + 1$

エ $n < M$

解答解説

流れ図の問題である。

左側のフローチャートの結果は、 $M \times (M - 1) \times \dots \times 2 \times 1$ となる。右側のフローチャートは、nのインクリメントが $x \times n$ の実行後であるから、Mまで乗じて、M+1になったときに終了すればよい。従って、終了条件は $n > M$ となる。求める答えはアとなる。

例題演習

n 個の要素を持つ配列中の値と探索すべきデータ X を順次比較し、配列中の値にデータ X が存在した場合、“有”を表示する。このとき、添字 $n+1$ の場所に探索すべき X を入れておく。

添字	1	2	...	i	...	n	$n+1$
値	a_1	a_2		a_i		a_n	X

この線形探索アルゴリズムの に入れるべき適切な条件はどれか。

- ステップ1 添字 i に 1 を入れる。
ステップ2 であればステップ5へとぶ。
ステップ3 添字 i に 1 を加算する。
ステップ4 ステップ2へとぶ。
ステップ5 添字 i が n 以下であれば“有”を表示する。
ステップ6 終了

- ア $i \geq n$ イ $i \neq n$ ウ $X = a_i$ エ $X \neq a_i$

解答解説

線形探索の番兵に関する問題である。

番兵は、探し求める値を配列の最後に付加して、配列の大きさ N の範囲内に求める値が見つければ探索成功、 $N+1$ で見つかったと探索不成功の判定をする線形探索法である。

番兵を利用した線形探索のアルゴリズムであり、繰返しからの脱出条件は配列の値 $= X$ である。答えは $X = a_i$ となり、求める答えはウとなる。

例題演習

次の記述の の中に入れる適当な語句はどれか。

昇順に整列済みの n 個のデータに対して二分探索を行う場合の探索終了条件は、探索成功または となっている。

- ア 小さいほうの要素番号 $\geq$ 大きいほうの要素番号
イ 小さいほうの要素番号 $>$ 大きいほうの要素番号
ウ 小さいほうの要素番号 $> n$ 、または、大きいほうの要素番号 < 1
エ 小さいほうの要素番号 $\geq n$ 、または、大きいほうの要素番号 ≤ 1

解答解説

2 分探索の探索終了条件に関する問題である。

2 分探索法の終了条件

- ① 探索データが見つかった場合
② 整列済みの配列に対して、探索範囲の上限の添字を H 、下限の添字を L とすると、 L

>Hになれば探索を終了する。
アの等号は探索が可能であり、終了条件にはならない。
ウ、エは探索範囲が変化するため最初の両端の1、nでは判定できない。
求める答えはイとなる。

例題演習

整列済みのデータを対象とした二分探索に関する次の記述の中で正しいものはどれか。

- ア データが昇順に整列されている場合、キー値の小さいデータの方がキー値の大きいデータより、少ない比較回数で探索できる。
- イ 検索するキーがデータの中にないと。無限プールに陥る。
- ウ データの要素数が2倍になると、キー値の最大比較回数は4倍になる。
- エ データ中に同じキー値をもつ要素が複数個ある場合、検索の結果発見できるのは、その中の特定の一つだけである。

解答解説

2分探索法の特徴に関する問題である。

アは、検索するキー値が小さい方になるか大きい方になるかは同じ確率で発生する。従って、キー値の小さいデータの値がキー値の大きいデータよりも少ない比較回数で探索できるとは限らない。

イの検索するデータが存在しない場合は、最後に探索対象の範囲内の要素数が0となり、二分探索が終了する。

ウの要素数が2倍になっても探索回数は1回増加する程度である。

エは、二分探索で比較されるのは、対象の範囲の中央の要素であるから、同じ内容の要素が複数存在しても、中央の要素になった一つが探索されるのみである。求める答えはエである。

例題演習

次の状態で記録されたファイルで、2分探索に最も適しているものはどれか。

- ア 磁気ディスク上に索引順編成で記録したファイル
- イ 磁気テープ上にレコードをハッシュ法によって記録したファイル
- ウ 主記憶上にレコードをキー順に記録したファイル
- エ 主記憶上にレコードを線形リストで記録したファイル

解答解説

2分探索に関する問題である。

2分探索の条件はキー値が昇順または降順に整列されていることである。

アの索引順編成はデータは入力順に並んでおり索引を利用して目的のデータを効率よく探索するものであるから、必ずしも2分探索には適さない。

イのハッシュ法によって記録されたデータはハッシュ関数の計算結果で格納場所が決められ

ており、2分探索可能なデータの並びになっていない。

ウはキー順に記録されたファイルであり、昇順または降順に整列したキーを利用して2分探索が可能である。求める答えはウである。

エの線形リストで記録されたデータは昇順または降順に並んでいるとは限らない。従って、2分探索には適さない。

例題演習

2分探索において、整列されているデータ数が4倍になると、最大探索回数はどうなるか。

ア 1回増える

イ 2回増える

ウ 約2倍になる

エ 約4倍になる

解答解説

2分探索のデータ数と探索回数の関係に関する問題である。

探索回数は $\log_2 N$ に比例する。

データ数を N とすると、最大探索回数は $\log_2 N + 1$ となる。データ数が4倍になると、最大探索回数は次のようになる。

$$\log_2 4N + 1 = \log_2 4 + \log_2 N + 1 = 2 + \log_2 N + 1 = 3 + \log_2 N$$

となる。従って、2回探索回数が増加する。求める答えはイとなる。

例題演習

データを降順に並べた線形リストを2分探索法で探索するとき、3回目までの比較で探索を終了することができる要素の最大個数はどれか。

ア 4

イ 5

ウ 7

エ 15

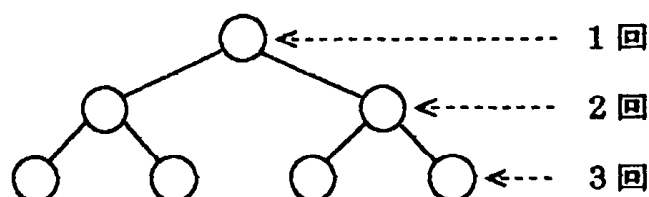
解答解説

2分探索におけるデータの個数と比較回数に関する問題である。

平均比較回数は $\log_2 N$ に比例する。2分探索木の考え方を利用して、高さが2の場合について、最大個数を求めればよい。

2分探索法では、中央値を比較し、探索値が中央値よりも大きい場合は左部分木へ、小さい場合には右部分木に進み、同様の比較を繰り返す。

図は、探索3回目までの2分木を表したものである。要素の最大個数は7個になる。求める答えはウとなる。



例題演習

次のような数値が格納された配列から数値 4 を二分探索法で検索する場合、比較回数はどれか。

2、4、6、7、8、10、11、12、15、16、18

ア 2

イ 3

ウ 4

エ 6

解答解説

2 分探索の比較回数を求める問題である。

添字	1	2	3	4	5	6	7	8	9	10	11
	2	4	6	7	8	10	11	12	15	16	18

1 回目の中央値は添字 6 で 10 と 4 との比較になる。次の範囲は 1 ～ 5 で、2 回目の中央値は添字 3 で 6 と 4 との比較になる。次の範囲は 1 ～ 2 で、3 回目の中央値は添字 1 で 2 と 4 の比較になる。4 回目の範囲は 2 ～ 2 で添字 2 の 4 と 4 の比較で一致する。求める答えはウとなる。

例題演習

ハッシュ法の説明として、適切なものはどれか。

- ア 関数を用いてレコードのキー値からレコードの格納アドレスを求めることによってアクセスする方法
- イ それぞれのレコードに格納されている次のレコードの格納アドレスを用いることによってアクセスする方法
- ウ レコードのキー値とレコードの格納アドレスの対応表を使ってアクセスする方法
- エ レコードのキー値をレコードの格納アドレスとして直接アクセスする方法

解答解説

ハッシュ法に関する問題である。

ハッシュ法はハッシュ関数を使用して、データの探索キーの値をデータの格納アドレスに変換することである。データの参照を高速で行う。ハッシュ法では、異なるキーから同じハッシュ値が求まる場合があり、これを衝突という。衝突した場合の対策として、チェーン法とオープンアドレス法がある。チェーン法は同じハッシュ値を持つデータをポインタを利用してリストでつなぐ方法である。オープンアドレス法は、元のハッシュ値 + 1 で再ハッシュを行う方法である。ハッシュ法はアドレスを計算で求めるため、計算量は通常 $O(1)$ である。従って、表探索に利用する場合も衝突が発生しなければ 1 回の計算で格納場所を決めることができる。

アの関数を用いてレコードのキー値からレコードの格納アドレスを求めるアクセス法はハッシュ法に関する内容である。求める答えはアとなる。

イのレコードに格納されている次のレコードの格納アドレスを用いる方法はリスト構造のデータのアクセスに使用する方法である。

ウのレコードのキー値と格納アドレスの対応表を使用する方法は索引編成の索引を用いるアクセス方式である。

エのレコードのキー値を格納アドレスとして使用する方法は直接編成の実アドレス指定方式である。

例題演習

キー x のハッシュ関数として $h(x) = \text{mod}(x, 97)$ を用いるとき、キー1094とハッシュ値が一致するものは、キー1～1000の中に幾つあるか。ここで、 $\text{mod}(x, 97)$ は x を97で割った余りを表す。

ア 9 イ 10 ウ 11 エ 12

解答解説

ハッシュ法に関する問題である。

キー1094のハッシュ値を求めると、 $\text{mod}(1094, 97) = 27$

キー1～1000の間で、ハッシュ値が27になるのは、 $97 \times 0 + 27$ 、 $97 \times 1 + 27$ 、…、 $97 \times 10 + 27$ の11個となる。求める答えはウとなる。

例題演習

16進数で表される9個のデータ1A, 35, 3B, 54, 8E, A1, AF, B2, B3を順にハッシュ表に入れる。ハッシュ値をハッシュ関数 $f(\text{データ}) = \text{mod}(\text{データ}, 8)$ で求めたとき、最初に衝突が起こる(既に表にあるデータと等しいハッシュ値になる)のはどのデータか。

ここで、 $\text{mod}(a, b)$ は a を b で割った余りを表す。

ア 54 イ A1 ウ B2 エ B3

解答解説

シノニムデータに関する問題である。

次の手順で求める。

- ① 9個の16進数のデータを10進数に変換する。
- ② ハッシュ関数を使用してハッシュ値を求める。
- ③ 最初にハッシュ値が一致するデータを求める。

16進数	1A	35	3B	54	8E	A1	AF	B2	B3
10進数	26	53	59	84	142	161	175	178	179
ハッシュ値	2	5	3	4	6	1	7	2	3

従って、最初にハッシュ値が一致するのはB2の2である。求める答えはウとなる。

(別解)

この問題ではハッシュ値を次のようにした求めることができる。

16進数を除数8で割る場合、16進数の2桁目より上位は必ず割り切れて余りは0となる。従って、ハッシュ値は16進数の1桁の値を8で割った余りである。1桁目の16進数が0～7までの場合はその値が余りになり、8～15までの場合は8を減じた数値になる。

例題演習

キー値の分布が1～1,000,000の範囲で一様ランダムであるデータ5個を、大きさ10のハッシュ表に登録する場合、衝突の起こる確率はおおよそ幾らか。ここで、ハッシュ値はキー値をハッシュ表の大きさで割った余りを用いる。

ア 0.2

イ 0.5

ウ 0.7

エ 0.9

解答解説

ハッシュ法の衝突に関する問題である。

5個の抽出されたデータを大きさ10のハッシュ表に重複を許して登録する場合の「場合の数」と重複を許さないで登録する場合の「場合の数」を求め、前者に対する後者の割合(衝突が発生しない割合)を求め、1から衝突しない割合を減ずると衝突する割合を求めることができる。

- ① 衝突が発生する場合を含めた5個のデータのハッシュ表に登録される場合の数は 10^5 通りである。
- ② 衝突しない場合の場合の数は、 $10 \times 9 \times 8 \times 7 \times 6 = 30240$ 通りになる。
- ③ 衝突しない確率は $30240 / 100000 = 0.3$
- ④ 衝突する確率は $1 - 0.3 = 0.7$ となる。

求める答えはウとなる。

例題演習

0000～4999のアドレスをもつハッシュ表があり、レコードのキー値からアドレスに変換するアルゴリズムとして基数変換法を用いる。キー値が55550のときのアドレスはどれか。ここで、基数変換法ではキー値を11進数と見なし、10進数に変換した後、下4けたに対して0.5を乗じた結果(小数点以下は切捨て)をレコードのアドレスとする。

ア 0260

イ 2525

ウ 2775

エ 4405

解答解説

ハッシュ法に関する問題である。

ハッシュ関数に基数変換法を用いる問題である。

基数変換法を用いて、11進数のキー値55550を10進数に変換する。

$$\begin{aligned} 5 \times (11^4 + 11^3 + 11^2 + 11) + 0 &= 5 \times (14641 + 1331 + 121 + 11) \\ &= 5 \times 16105 = 70520 \end{aligned}$$

求めるアドレスは $70520 \times 0.5 = 35260$ となり、求める答えはアとなる。

txt010222 整列アルゴリズム

① 整列のアルゴリズム

① 整列とは

整列はデータ列をある規則に従って並べ替えることである。小さい順に並び替えることを昇順に整列といい、大きい順に並び替えることを降順に整列という。

② 内部整列と外部整列

内部整列は対象とするデータ群が主記憶装置にある場合の整列である。外部整列は対象とするデータ群が補助記憶装置にある場合の整列である。

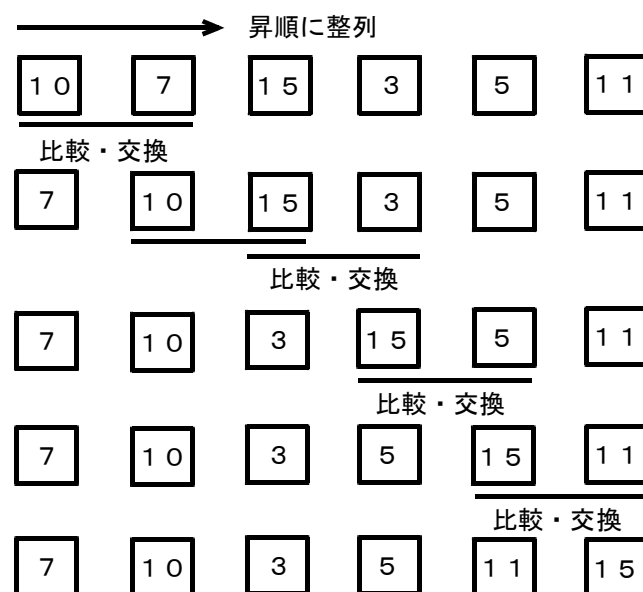
③ 整列の計算量

データを並べ替える場合、それぞれのアルゴリズムの考え方に基づいて、データの比較が行われる。この比較の回数を計算量という。計算量の小さいアルゴリズムがCPUを効率的に使用するアルゴリズムになる。アルゴリズムは常に計算量を問題にする。

② 基本交換法 (バブルソート)

① 基本交換法

基本交換法は配列の隣り合った二つのデータを比較して、大小の順序が違っていると並べ替えていく方法でデータを整列する方法である。



⑥ 基本交換法の手順

- ㊦ 配列の先頭のデータと次のデータの大小を比較する。大小関係が異なると入れ替える。
- ㊧ 2番目のデータと次のデータの大小を比較する。大小関係が異なると入れ替える。
- ㊨ 順次、移動して、隣り合う2つのデータを比較しながら、大小関係が異なると入れ替える操作を繰り返す。
- ㊩ 配列の最後まで来たら1回目の比較を終了する。この1回目の比較で、最後の要素の位置が決まる。
- ㊪ 2回目も同様の比較を行い、2回目の比較の中で最後方の要素の位置が決まる。
- ㊫ 各回の操作で、その回の最後方の要素の位置が決まる。
- ㊬ すべての配列の要素が昇順または降順に並ぶと操作を完了する。

⑦ 基本交換法の比較回数

対象のデータが n 個ある場合、1回の操作で $n - 1$ 回の比較が行われ、1つのデータの正しい位置が決まる。全部のデータの位置を決めるのに必要な比較回数は

$$(n - 1) + (n - 2) + \cdots + 2 + 1 = n(n - 1) / 2 = (n^2 - n) / 2 \approx n^2$$

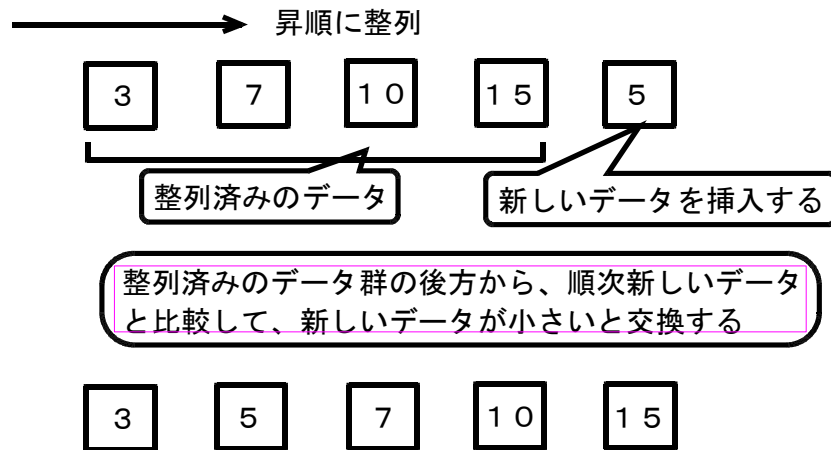
となる。基本交換法の比較回数は n^2 に比例する。

③ 基本挿入法(挿入ソート)

㊐ 基本挿入法

基本挿入法は整列済みのデータ系列に対して、データを1件ずつ正規の位置に挿入し、その位置以降のデータを順次後方に移動させて、並べ替える方法である。具体的な新しいデータの挿入位置の決定は手順は、昇順に整列する場合、次のようにして行う。

- ㊦ 新しく挿入するデータを列の最後方の位置に仮置きする。
- ㊧ 列の後方から挿入したデータと既存のデータを比較して、挿入データが小さい場合には前方に移動して一つ前のデータと比較する。
- ㊨ 順次前方に向かって調べ、挿入データが大きい場合や等しい場合にその位置に挿入する。



⑥ 基本挿入法の手順

- ㊦ 整列済みのデータの最後方の要素の次に新しいデータを位置づける。
- ㊧ 新しいデータと整列済みの最後方のデータを比較して、大小関係が異なると、二つのデータを交換する。
- ㊨ 交換すると、整列済みの次の前のデータと新しいデータを比較して、大小関係が異なると、二つのデータを入れ替える。
- ㊩ 比較する二つのデータが異なる間、㊧～㊨の操作を繰り返す。
- ㊪ 大小関係が正しくなると、その位置が新しいデータの挿入位置になる。
- ㊫ 新しいデータがなくなるまで、㊦～㊪の操作を繰り返す。

⑦ 基本挿入法の比較回数

基本挿入法の比較回数は n^2 に比例する。

④ 基本選択法(選択ソート)

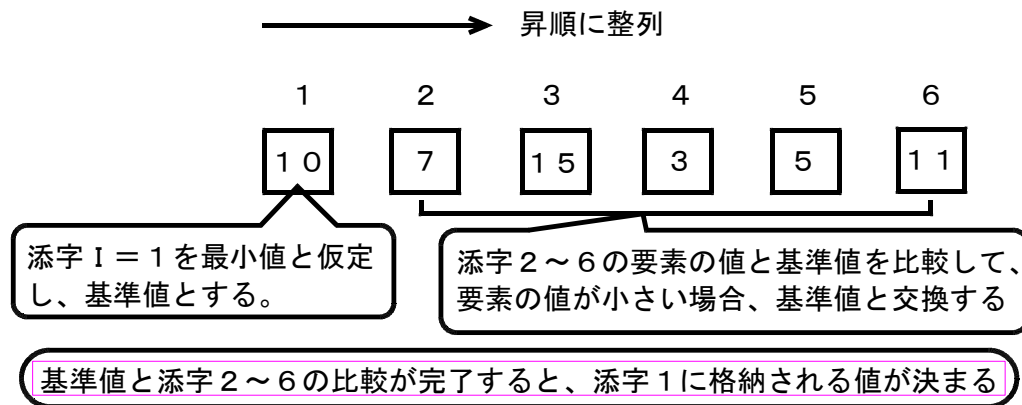
㊱ 基本選択法

基本選択法は隣接データの並べ替えを行わず、各回の比較で対象データの最大値(最小値)を求め、最終格納位置のデータと交換する整列法である。

昇順に整列する場合、最も左の要素をデータ格納位置と決めると、その要素の値と右側の各

要素の値を順次比較して格納位置の要素の値が大きい場合には交換し、小さい値の場合はそのままにして次の右側の値と比較して同様の処理を行う。右側の各要素を最後まで比較してデータの格納位置の値を決める。

次の格納位置を左から2番目として、同様の比較を行い格納する値を決める。順次、残りのデータの最小値と格納位置のデータを比較し、格納位置のデータが大きい場合に最小のデータと交換する操作を繰り返す整列法である。



⑥ 基本選択法の手順

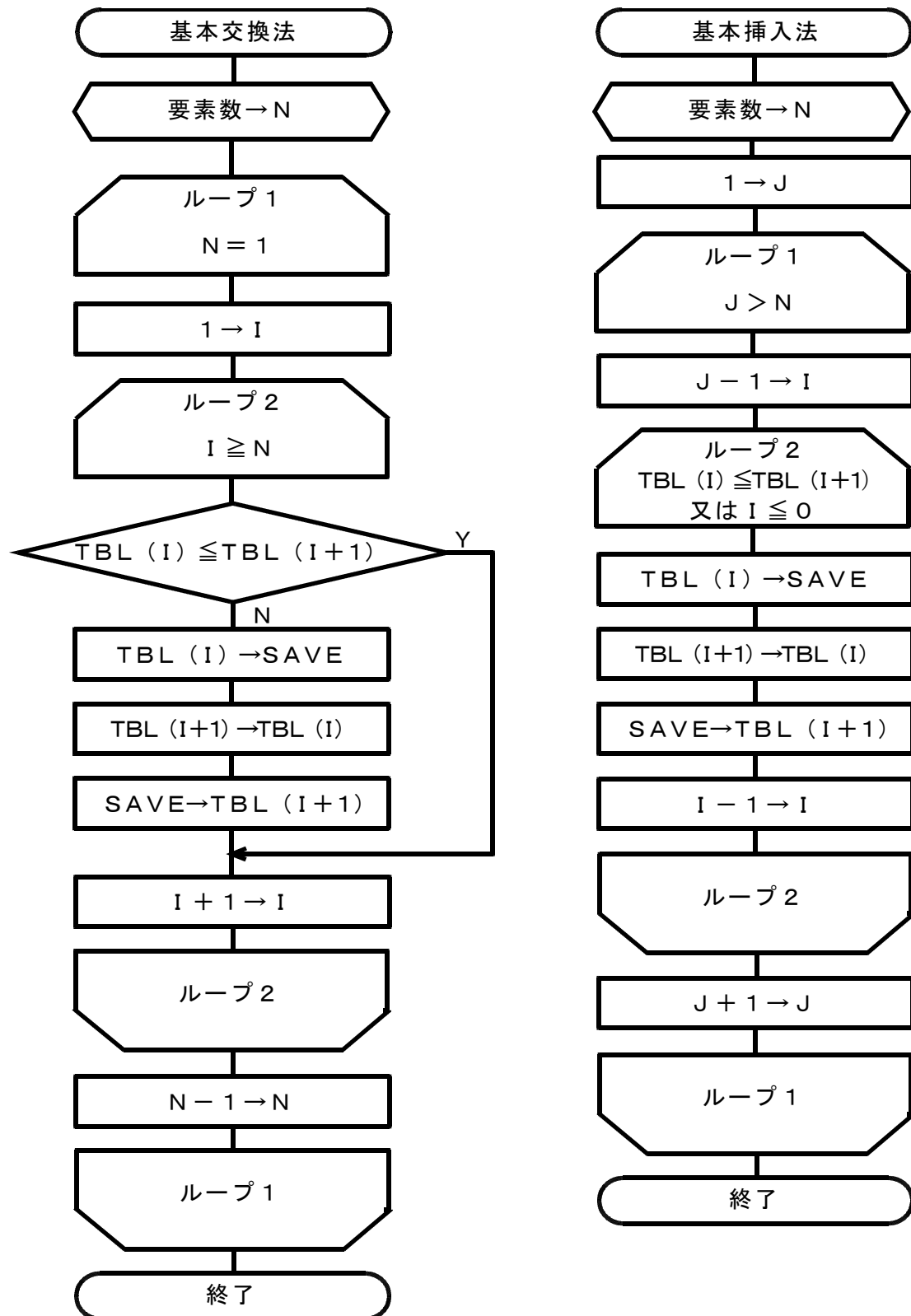
- ㊦ 配列の先頭の要素を決定対象の要素として選択する。
- ㊧ 決定対象の要素とその他の要素のデータを順次比較して、大小関係が異なると、データを交換する。
- ㊨ 配列の最後まで比較すると、1回目の比較を終了する。この比較で先頭の要素の値が決まる。
- ㊩ 配列の残りの要素の先頭の要素を決定対象に選択し、㊧、㊨の操作を繰り返す。この比較で配列の2番目の要素の値が決まる。
- ㊪ 順次、残りの配列の先頭の要素を決定対象に選択し、㊧、㊨の操作を繰り返す。
- ㊫ 配列の最後の一つ前の要素の値が決まると整列の操作が完了する。

⑦ 基本選択法の比較回数

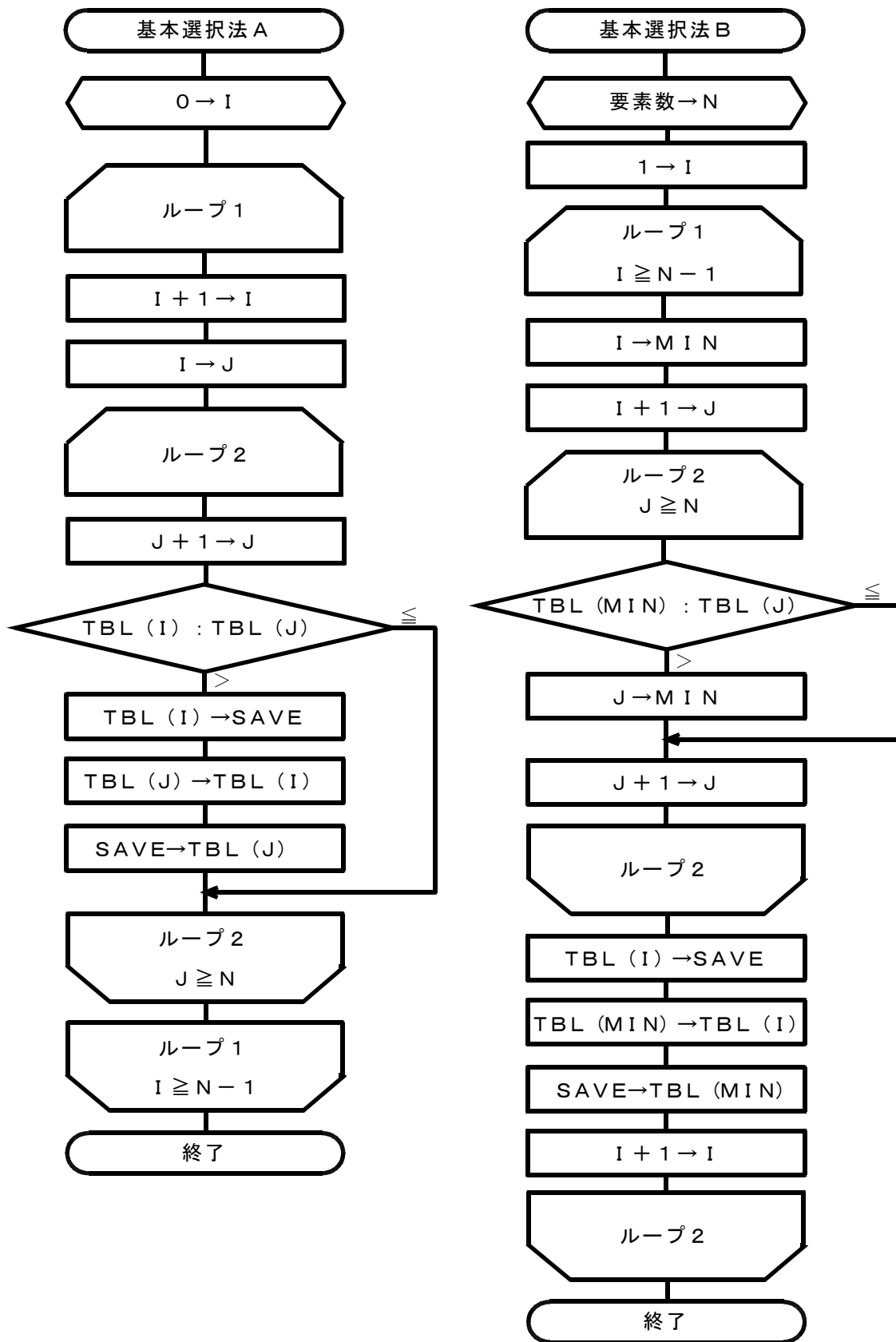
基本選択法の比較回数は n^2 に比例する。

⑤ 基本整列アルゴリズムの流れ図

① 基本交換法・基本挿入法の流れ図



⑥ 基本選択法の流れ図



例題演習

データ列の隣り合う要素の値を比較し、小さい方が右にあれば交換する。この操作をデータ列の左端から右端まで繰り返す処理を1回のパスとする。次のデータ列でパスを2回繰り返した後のデータ列の内容を示しているものはどれか。

5	4	1	3	6	2
---	---	---	---	---	---

ア

1	3	2	4	5	6
---	---	---	---	---	---

イ

1	3	4	2	5	6
---	---	---	---	---	---

ウ

4	1	5	3	2	6
---	---	---	---	---	---

エ

4	1	5	3	6	2
---	---	---	---	---	---

解答解説

バブルソートの並べ替えに関する問題である。

並べ替えは次のようになる。

① 1回目の並べ替え

5 4 1 3 6 2
4 5 1 3 6 2
4 1 5 3 6 2
4 1 3 5 6 2
4 1 3 5 2 6

② 2回目の並べ替え

4 1 3 5 2 6
1 4 3 5 2 6
1 3 4 5 2 6
1 3 4 2 5 6

2回目の並べ替えを完了した時点での並びは 1 3 4 2 5 6の順であり、求める答えはイとなる。

例題演習

図はある配列をソートしたときに要素の順序が変わっていく様子である。このソートのアルゴリズムは、次のうちのどれか。

(初期状態) 4、3、7、6、2、1、5
3、4、7、6、2、1、5
3、4、6、7、2、1、5
2、3、4、6、7、1、5
1、2、3、4、6、7、5
(整列後) 1、2、3、4、5、6、7

ア クイックソート

イ 選択ソート

ウ 挿入ソート

エ バブルソート

解答解説

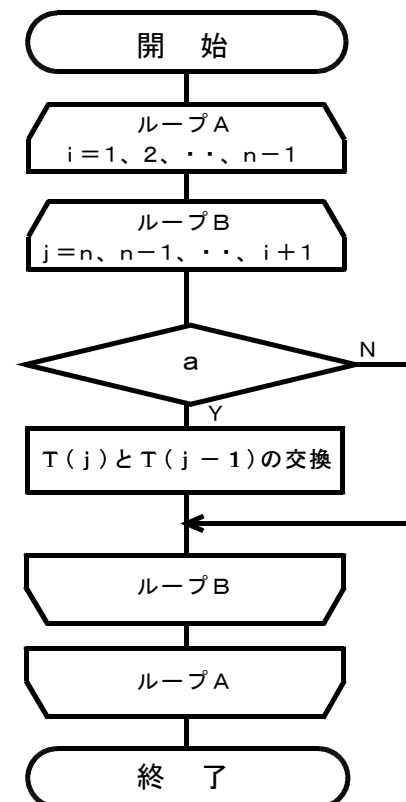
挿入ソートの操作手順に関する問題である。

初期状態で整列済みのデータは4のみである。1回目に3が対象になって4の前に設定される。2回目は整列済みの3, 4に7を挿入する。この場合は交換する必要がない。3回目は整列済みの3, 4, 7に6を挿入する場合で、6は4と7の間に挿入される。4回目は2を挿入する場合で、7と2、6と2、4と2、3と2を順次交換し、整列済みの先頭に挿入される。5回目は1を挿入する場合で、7と1、6と1、4と1、3と1、2と1を順次交換し、整列済みの先頭に挿入される。6回目は5を挿入する場合で4と5の間に挿入される。以上の操作は挿入ソートで、求める答えはウとなる。

例題演習

整数値からなる n 個（ただし、 $n \geq 2$ ）のデータが、配列 T に格納されている。次の流れ図は、それらのデータを交換法を用いて昇順に整列する処理を示す。流れ図中の a に入れる適切な条件はどれか。

- ア $T(j) < T(j+1)$
- イ $T(j) < T(j-1)$
- ウ $T(j) > T(j+1)$
- エ $T(j) > T(j+1)$



解答解説

基本交換法の流れ図に関する問題である。

昇順に整列する基本交換法の手順

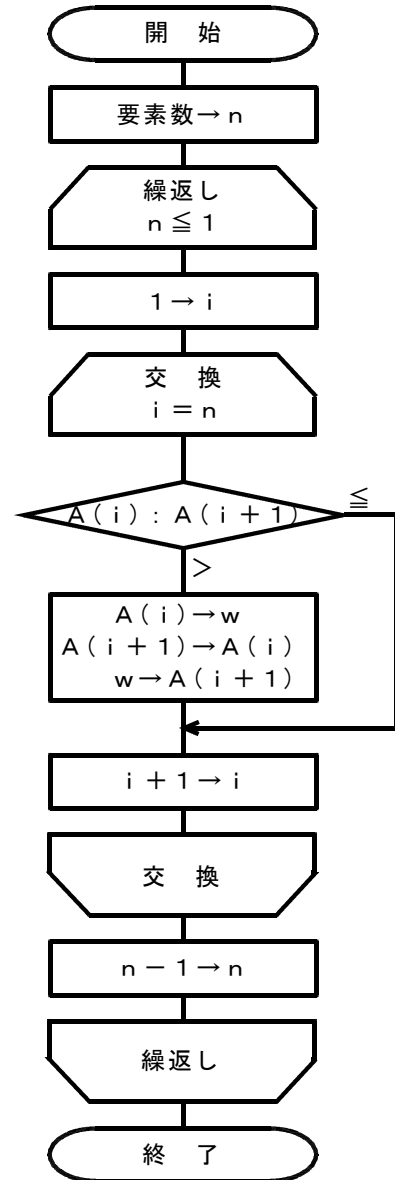
- ① 隣り合った数値を比較して大小関係が異なっていると交換する。
- ② 全データの一通りの大小の比較で最も大きいデータの整列位置が決まる。
- ③ 残りのデータについて同様の比較を繰り返す。
- ④ 全比較回数はデータ数の2乗に比例する。
- ⑤ 流れ図は二重ループになる。

ループBが1回の一通りのデータの比較に相当する。流れ図では、データの比較は後方から前方に向かって行われており($j = n, n-1, \dots, 2, 1$)、後のもの($T(j)$)と一つ前のもの($T(j-1)$)を比較し、 $T(j) < T(j-1)$ ならば交換し、そうでなければ何もしない方式になっている。従って、aの内容は $T(j) < T(j-1)$ で、求める答えはイとなる。

例題演習

右の流れ図が表す整列アルゴリズムはどれか。

- ア クイックソート
- イ シェルソート
- ウ 挿入ソート
- エ バブルソート



解答解説

流れ図の処理方式から、整列アルゴリズムを判定する問題である。

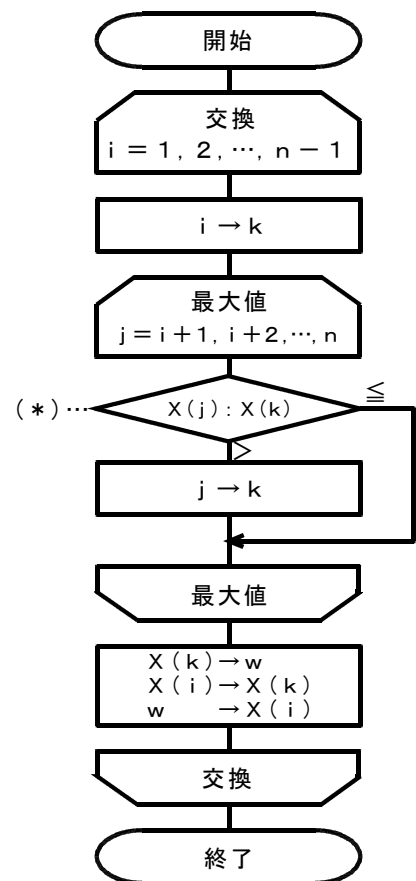
この流れ図は、 $i = 1$ から隣り合う配列の大小を比較して、 $A(i) > A(i+1)$ ならば交換し、全要素の数 n を比較して最大の要素一つを決定する。再び、残りの $n-1$ 個に対して同じ操作を繰り返して、 $n \leq 1$ になるまで $n-1$ 回の操作を繰り返す。

この方法によって配列を昇順に整列するアルゴリズムで、バブルソートである。求める答えはエとなる。

例題演習

次の流れ図は、最大値選択法によって値を大きい順に整列するものである。*印の処理（比較）が実行される回数を表す式はどれか。

- ア $n - 1$
- イ $n(n - 1) / 2$
- ウ $n(n + 1) / 2$
- エ n^2



解答解説

基本選択法に関する問題である。

基本選択法の手順（降順に整列する場合）

- ① 配列中の要素を左から順次比較することにより、配列内の最大値を求める。
- ② その最大値が格納されている要素と左端の要素を入れ替え、左端に最大値を求める。
- ③ 左端の位置を右に一つずらし、再度左端から順次比較し最大値を求める。
- ④ 以上の処理を要素数 $N - 1$ 回繰り返し、昇順に整列される。

流れ図の最大値の繰り返し

1回の最大値を決めるための操作であり、この最大値の繰り返しが次に示すように $n - 1$ 回繰り返されることになる。

流れ図の交換の繰り返し

1回目の最大値を求める場合の比較回数は、データ数を n とすると、 $n - 1$ 回となる。

2回目の最大値を求める場合の比較回数は、データ数が $n - 1$ で、 $n - 2$ 回となる。

3回目の最大値を求める場合の比較回数は、データ数が $n - 2$ で、 $n - 3$ 回となる。

同様にして

$n - 2$ 回目の最大値を求める場合の比較回数は、データ数が 3 で、 2 回となる。

$n - 1$ 回目の最大値を求める場合の比較回数は、データ数が 2 で、 1 回となる。

比較回数の総数

以上の $n - 1$ 回の最大値の繰り返しの比較回数の総数 S は次のようになる。

$$S = (n - 1) + (n - 2) + (n - 3) + \cdots + 3 + 2 + 1$$

次のようにして計算する。

$$S = (n-1) + (n-2) + (n-3) + \cdots + 3 + 2 + 1 \quad \cdots \cdots ①$$

$$S = 1 + 2 + 3 + \cdots + (n-3) + (n-2) + (n-1) \quad \cdots \cdots ②$$

①+②を求めると

$$2S = n + n + n + \cdots + n + n + n \quad (n \text{ が } n-1 \text{ 回加算される})$$

$$= n(n-1)$$

$$S = n(n-1)/2$$

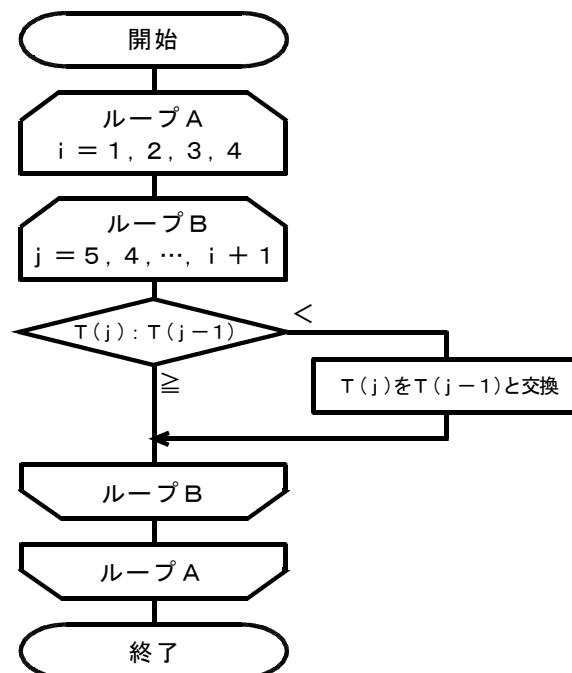
となり、求める答えはイとなる。

例題演習

次の流れ図で示されるアルゴリズムで配列 T を整列する。i = 1 でループ B が終わったときの配列 T の内容として、正しいものはどれか。

配列 T の初期値

T(1)	31
T(2)	2
T(3)	24
T(4)	15
T(5)	40



ア	2	イ	2	ウ	31	エ	40
	24		31		24		31
	15		15		15		2
	31		24		40		24
	40		40		2		15

解答解説

バブルソートの流れ図に関する問題である。

ソートする配列 31 2 24 15 40 に対して規則を用いると次のようになる。

31 2 24 15 40

比較

31 2 24 15 40

比較・交換

31 2 15 24 40

比較

31 2 15 24 40

比較・交換

2 31 15 24 40 $i = 1$ の操作が終了する。

求める答えはイとなる。

例題演習

配列 $A[i]$ ($i = 1, 2, \dots, n$) を、次のアルゴリズムによって整列する。行 2 ～ 3 の処理が初めて終了したとき、必ず実現されている配列の状態はどれか。

〔アルゴリズム〕

行番号

- 1 i を 1 から $n - 1$ まで 1 ずつ増やしながら行 2 ～ 3 を繰り返す
- 2 j を n から $i + 1$ まで 1 ずつ減らしながら行 3 を繰り返す
- 3 もし $A[j] < A[j - 1]$ ならば、 $A[j]$ と $A[j - 1]$ を交換する

ア $A[1]$ が最小値になる。

イ $A[1]$ が最大値になる。

ウ $A[n]$ が最小値になる。

エ $A[n]$ が最大値になる。

解答解説

整列のアルゴリズムに関する問題である。

後方からバブルソートで昇順に整列するアルゴリズムである。この場合の比較は、基準値 $A[j]$ と一つ前の要素 $A[j - 1]$ と比較して、基準値が小さい場合に交換する。前の要素が大きく、後ろの要素が小さい場合に交換する。前に小さな値がくる整列である。

例えば、2、4、1、5、3 の数列について考える。

$i = 1$ の場合、2、4、1、5、3 → 2、4、1、3、5 → 2、1、4、3、5
→ 1、2、4、3、5 となる。

$i = 2$ の場合、2、4、3、5 の数列を $i = 1$ の場合と同様にして、5 から 3 までを比較して交換し、最小値 2 を求める。

以下、 $i = 4$ まで、同様の操作を繰り返して昇順に整列する。

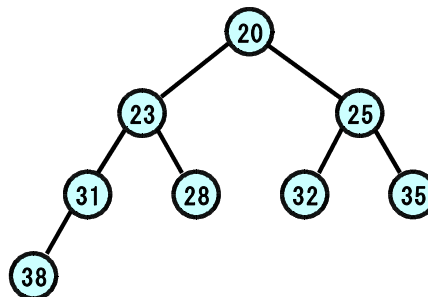
従って、 $A[1]$ が最小値になる。求める答えはアとなる。

① ヒープソート

① ヒープソートとは

ヒープソートは、ソートすべきデータをヒープ構造の木を利用して行う方法である。ヒープソートは、根の部分のデータを取り出した後、ヒープを再構成する作業を繰り返せば、ソート済みデータ系列が得られる。ヒープ木は完全二分木のデータ構造を配列で表現したもので、どの親子の値をとってみても必ず「親の値<子の値」(または「親の値>子の値」)の関係になっている配列である。親の節の値が子の節の値より小さくなるようにすれば、根の値は二分木の中で最小になる。逆にすれば最大になる。

ヒープ木の例
親の値<子の値

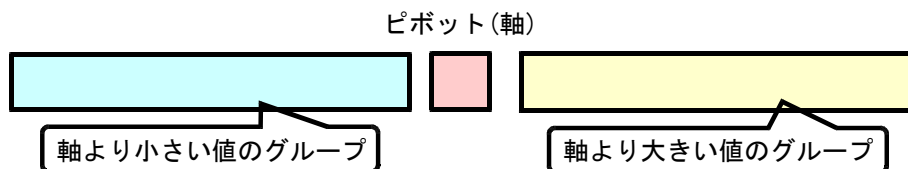


② ヒープソートの比較回数

平均比較回数も最大比較回数も $n \log_2 n$ となる。

② クイックソート

① クイックソートとは



適当にサンプリングして得られたデータ N に着目し、 N より小さいデータ系列と N 以上のデータ系列に 2 分割する。このデータ N を軸 (ピボット) という。

データ N を軸として、その軸の値より小さい要素は軸より前方へ、大きな要素は軸より後方へ振り分ける。適当な長さの系列になるまで分割を繰り返し、それぞれの系列をソートすれば 1 つのソート済み系列となる。

分割の繰返しに再帰呼び出しを利用した高速の整列法である。ピボットの選び方には、中央

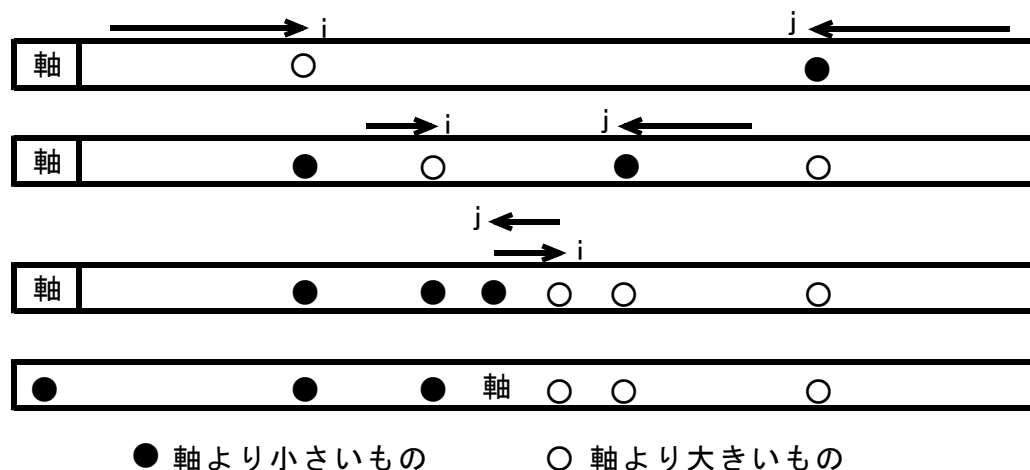
値や3要素(右・中央・左)の中間値の考え方がある。

⑥ クイックソートのアルゴリズム

数列の左側から走査していく変数を i 、右側から走査していく変数を j とすると、最左端の基準値に対して、左部分列と右部分列の走査は次のようになる。

- ㊦ 要素番号 i を数列の最左端 + 1、要素番号 j を数列の右端に設定する。
- ㊧ 数列を左から右に走査して、要素の値が基準値以上になる要素番号の位置 i を見つける。
- ㊨ 数列を右から左に走査して、要素の値が基準値より小さくなる要素番号の位置 j を見つける。
- ㊩ 要素番号 $i \geq$ 要素番号 j ならば、ループを抜け、㊦に進む。
- ㊪ 要素番号 $i <$ 要素番号 j ならば、要素番号 i 項と要素番号 j 項を交換し、㊧に戻る。
- ㊫ 基準値と要素番号 j 項を交換する。

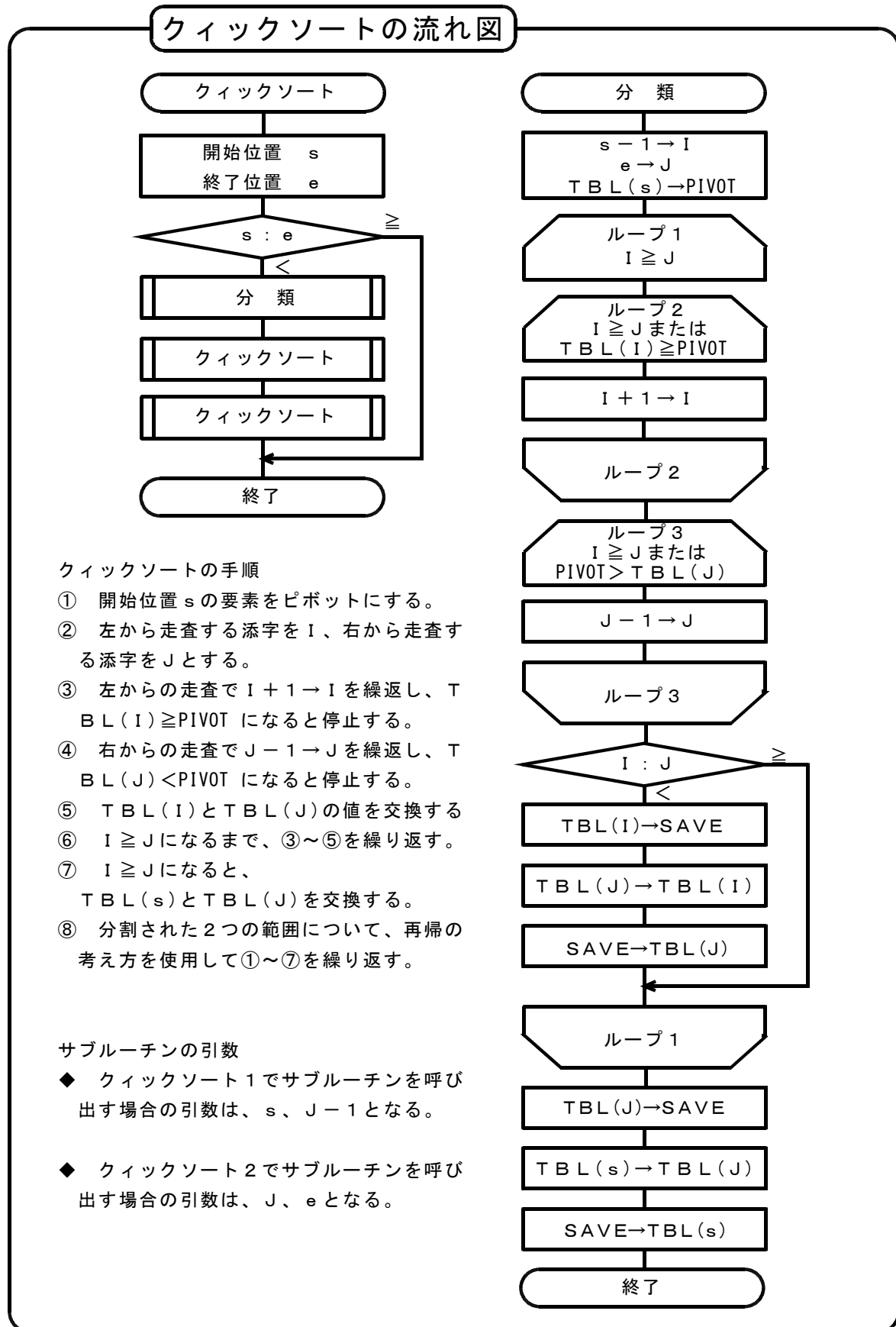
次の対象になる左部分列は左端から $j - 1$ 、右部分列は j から右端の2つの範囲について同様の操作を繰り返す。



⑦ クイックソートの比較回数

平均比較回数は $n \log_2 n$ 。最大比較回数は n^2 となる。比較回数の求め方は、クイックソートの計算量の求め方を参照のこと。

④ クイックソートの流れ図



③ クイックソートの具体例

次に示す10個のデータをクイックソートのアルゴリズムを利用して整列する。

41 24 76 11 45 64 21 69 19 36

操作の手順

㉞ 基準値を41に設定し、左側の初期値を24、右側の初期値を36に設定する。

㉟ 左側からの走査で基準値41以上の値は76、右側の走査から41以下の値は36となる。36と76を入れ替える。

41 24 36 11 45 64 21 69 19 76

㊱ 左側の走査で基準値以上は45、右側の走査から41以下の値は19となる。

41 24 36 11 19 64 21 69 45 76

㊲ 左側の走査で基準値以上は64、右側の走査から41以下の値は21となる。

1 2 3 4 5 6 7 8 9 10

41 24 36 11 19 21 64 69 45 76

㊳ 左側の走査で基準値以上は64、右側の走査から41以下の値は21となり、64の要素番号は7、21の要素番号は6となり、 $7 > 6$ となる。

㊴ 左側の41と21を入れ替えると、

21、24、36、11、19、41、64、69、45、76

41を基準値にして左部分列と右部分列が求められたことになる。次には、21～41の左部分列のグループと64～76の右部分列のグループの2つのグループに分けて、同様の処理を繰り返す。

③ シェルソート

㉠ シェルソートとは

データ列の中からX間隔ずつ離れたデータを取り出して、その部分列のデータを挿入ソートの考え方を利用して整列する。このX間隔のギャップを、始めは大きく、次第に小さくして最後に1にする。ギャップの決め方は、データ数の $1/2$ 、 $1/4$ 、…、1とする方式を用いる。例えば、データが8個ある場合、最初の飛びは $8/2 = 4$ とし、飛びの対のデータについて挿入ソートを使用する。次に、 $4/2 = 2$ の飛びの対について挿入ソートを使用し、最後に $2/2 = 1$ の飛び、通常の挿入ソートを行う。挿入ソートは、常に隣の要素と比較・交換するため、実行速度が遅いという欠点があったが、それを改善し、順番の違いを早く修正することができる。

⑥ シェルソートの比較回数

比較回数は $n^{1.5}$ に比例する。

④ シェーカーソート

① シェーカーソートとは

昇順に整列する場合、バブルソートは常に左から右に比較していき、右端に最大値がくるように整列する。シェーカーソートは左から右へ比較していき、右端に最大値を固定した後に、逆に右から左へ比較していき、最小値を左端に固定する方式で、この操作を繰り返してデータを昇順に並べ替える方式である。左から右に比較していく場合、データを交換した最後の位置を認識し、その位置を起点にして右から左への比較を開始する。右から左に比較する場合も、データを交換した最後の位置を認識し、その位置を起点として左から右への比較を開始する。

② シェーカーソートの具体例

最初は左から右に走査し、最後に交換した13の位置を記憶する。

19	10	26	43	21	13	比較回数左から 5 回
↑					↑	
10	19	26	43	21	13	
10	19	26	43	21	13	
10	19	26	43	21	13	
10	19	26	21	43	13	
10	19	26	21	13	43	
				↑		

次に13の位置を出発点にして、右から左に走査し、最後に交換した19の位置を記憶する。

10	19	26	21	13	43	比較回数右から 4 回
↑				↑		
10	19	26	13	21	43	
10	19	13	26	21	43	
10	13	19	26	21	43	
		↑		↑		

その次の走査は、19の位置と13と21を交換した21の位置との間で同じ走査を繰り返し、最後に交換した位置を記憶する。走査の左の起点 L と右の起点 R が $L \geq R$ となると、走査を終了する。

⑤ バブルソートの計算量

① バブルソートの計算量の求め方

この方法は比較回数は多いが、特別な作業領域は必要としない特徴がある。

操作手順は左端から順番に隣接要素を比較する。左側の要素＞右側の要素の時、左右の要素を交換する。この一連の処理で最大の要素が右端に位置する。

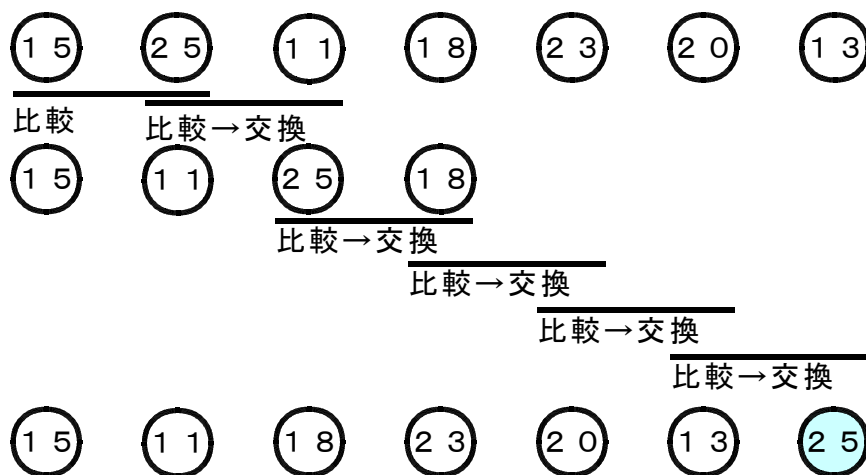
次に残りの要素に対して、再度左端から比較を行い最大値を右端に置く。

以上の処理を $n - 1$ 回繰り返すと昇順に整列する。この場合の計算量はデータ数を n とすると、 n^2 に比例する。

要素数 n とすると、比較回数 M は次の計算法で求まる。

$$\begin{aligned} M &= (n - 1) + (n - 2) + \cdots + 2 + 1 \\ &= (n - 1) \times \{(n - 1) + 1\} / 2 \\ &= (n^2 - n) / 2 \div n^2 / 2 \text{ (回)} \end{aligned}$$

② バブルソートの操作手順例と計算量



図の交換操作で25の位置が決定する手順は次の通りとなる。

- ㊦ 15と25を比較する。昇順であり、交換必要なし。
- ㊧ 25と11を比較する。大小が逆で交換する。
- ㊨ 25と18を比較する。大小が逆で交換する。
- ㊩ 25と23を比較する。大小が逆で交換する。
- ㊪ 25と20を比較する。大小が逆で交換する。

㊦ 25と13を比較する。大小が逆で交換する。

㊧ 25の位置が決定する。

この場合の比較回数は6回である。25を除いた残りの6要素について、同様に比較・交換を行い、23の位置を決定する。この2回目の比較回数は5回となる。

以下同様の操作を実行すると、比較回数は、4回、3回、2回、1回となる。従って、全比較回数は $6 + 5 + 4 + 3 + 2 + 1 = 21$ (回)となる。

⑥ クイックソートの計算量

㊐ クイックソートの操作

整列対象の要素数を n とし、ある基準値を設定し、その基準値より大きい要素のグループと小さい要素のグループに分ける。次の段階では2つのグループにそれぞれの基準値を設定し、それぞれの基準値より大きい要素のグループを2グループ、小さい要素のグループを2グループ作成する。次の段階では4グループに対して同じ操作を繰り返す。各グループの要素数が1になるまで繰り返す回数は $\log_2 n$ となり、1回の比較回数は n 回となる。

㊑ クイックソートの計算量の求め方

全体の個数 n から各グループの要素が1になるまでの段階数 k は次のようにして求める。

段階数が最小となるのは各段階毎にデータが2等分される場合であるから、比較回数を k とすると、次式が成立する。

$$2^k \leq n < 2^{k+1}$$

2を底とする対数をとると

$$\log_2 2^k \leq \log_2 n < \log_2 2^{k+1}$$

$$k \leq \log_2 n < k + 1$$

故に、 $k = \lceil \log_2 n \rceil$ となり、比較回数は $n \times \lceil \log_2 n \rceil$ となる。

比較数が最大となるのは、各段階で基準値によって振り分けられる場合で、この場合の段階数は n となり、比較回数の合計は n^2 に比例することになる。クイックソートの計算量は、データ数を n として平均比較回数で $n \times \lceil \log_2 n \rceil$ 、最大比較回数で n^2 となる。

例題演習

あるコンピュータで1,000個のデータをバブルソートを用いて整列するのに、1秒かかった。同様のデータ100,000個ではおよそ何秒かかるか。

但し、バブルソートの手間はデータ数の2乗に比例する。

ア 10

イ 100

ウ 1,000

エ 10,000

解答解説

バブルソートの計算量に関する問題である。

バブルソートの計算量は、データ数の2乗に比例するため、1000個のデータの操作時間が1秒であるから、100000個の場合の操作時間をTとすると、次の式が成り立つ。

$$(10^3)^2 : (10^5)^2 = 1 : T$$

$$10^6 : 10^{10} = 1 : T$$

$$1 : 10^4 = 1 : T$$

従って、 $T=10000$ (秒)となる。求める答えはエである。

例題演習

クイックソートの処理方法を説明したものはどれか。

- ア 既に整列済みのデータ列の正しい位置に、データを追加する操作を繰り返していく方法である。
- イ データ中の最小値を求め、次にそれを除いた部分の中から最小値を求める。この操作を繰り返していく方法である。
- ウ 適当な基準値を選び、それより小さな値のグループと大きな値のグループにデータを分割する。同様にして、グループの中で基準値を選び、それぞれのグループを分割する。この操作を繰り返していく方法である。
- エ 隣り合ったデータの比較と入替えを繰り返すことによって、小さな値のデータを次第に端の方に移していく方法である。

解答解説

整列アルゴリズムのクイックソートに関する問題である。

クイックソートは、軸となるデータに着目し、そのデータより大きい系列と小さい系列に分類する。新しくできた二つの系列に対してそれぞれ軸となるデータを設定し、大きい系列と小さい系列に分類する。この操作を繰り返しながら対象のデータを昇順または降順に並べ替える方法である。操作の展開には再帰の考え方が用いられる。

アは基本挿入法、イは基本選択法、ウはクイックソート、エは基本交換法である。求める答えはウとなる。

例題演習

2,000 個の相異なる要素が、キーの昇順に整列された表がある。外部から入力したキーによってこの表を2分探索して、該当するキーの要素を取り出す。このときのキーの比較回数は最大何回か。ただし、該当するキーは必ず表中にあるものとする。

ア 10

イ 11

ウ 12

エ 13

解答解説

探索データが存在する場合の2分探索法の最大比較回数を求める問題である。

最大比較回数を k とすると、要素数 N との間に次の条件が成立する。

$$2^{k-1} \leq N < 2^k$$

$N=2000$ として、 k の値を求める。

$$2^{k-1} \leq 2 \times 10^3 < 2^k$$

$$2^{k-1} \leq 2 \times 10^3 \text{ から}$$

$$k-1 \leq \log_2 2 + \log_2 10^3$$

$$k \leq 1 + 1 + 3 \div \log_{10} 2 = 2 + 3 \div 0.301 = 2 + 9.967 = 11.967 \dots \textcircled{1}$$

$$2 \times 10^3 < 2^k$$

$$k > \log_2 2 + \log_2 10^3 = 1 + 3 \div 0.301 = 1 + 9.967 = 10.967 \dots \textcircled{2}$$

①、②の式から $k=11$ となる。求める答えはイとなる。

例題演習

次の手順はシェルソートによる整列を示している。データ列“7, 2, 8, 3, 1, 9, 4, 5, 6”を手順(1)～(4)に従って整列すると、手順(3)を何回繰り返して完了するか。ここで、[]は小数点以下を切り捨てる。

[手順]

- (1) [データ数 $\div$ 3] $\rightarrow$ H とする。
- (2) データ列を互いに H 要素分だけ離れた要素の集まりからなる部分列とし、それぞれの部分列を挿入法を用いて整列する。
- (3) [H $\div$ 3] $\rightarrow$ H とする。
- (4) H が 0 であればデータ列の整列は完了し、0 でなければ(2)に戻る。

ア 2

イ 3

ウ 4

エ 5

解答解説

シェルソートに関する問題である。

シェルソートは、挿入ソートを改良したもので、データ列の中から X 間隔ずつ離れたデータを取り出して、その部分列のデータを挿入ソートの考え方を利用して整列する。この X 間隔のギャップを、始めは大きく、次第に小さくして最後に 1 にする。ギャップの決め方はいろいろあるが、よく用いられる方法にデータ数の $1/2$ 、 $1/4$ 、 $\dots$ 、1 とする方式を用いる。

データ数が 9 であるから、 $9 \div 3 \rightarrow H=3$ 、従って、3 だけ離れた要素を挿入法で整列する。

$$3 \quad 1 \quad 6 \quad 4 \quad 2 \quad 8 \quad 7 \quad 5 \quad 9$$

$3 \div 3 \rightarrow H=1$ となり、

$$1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9$$

$1 \div 3 \rightarrow H=0$

手順 3 の実行回数は 2 回となる。求める答えはアとなる。

例題演習

6 個の数値 180, 315, 282, 410, 645, 525 を並べ替える。手順 1 ～ 4 は途中までの手順を示したものである。手順 4 まで終わったときの結果はどれか。

手順 1 並びの左側から順に、数値の 1 の位の値によって 0 ～ 9 のグループに分ける。

手順 2 次に 0 のグループの数値を左側から順に取り出して並べ、その右側に 1 のグループ、以下順に 2 ～ 9 のグループの数値を並べていく。

手順 3 手順 2 で得られた数値の並びの左側から順に、数値の 10 の位によって 0 ～ 9 のグループに分ける。

手順 4 手順 2 と同様に、0 のグループの数値から順に並べる。

ここで、グループ内では、処理が行われた数値を左側から順に並べるものとする。

ア 180, 282, 315, 410, 525, 645

イ 315, 410, 525, 180, 282, 645

ウ 410, 315, 525, 645, 180, 282

エ 645, 525, 410, 315, 282, 180

解答解説

整列に関する問題である。

① 1 の位の値によって、0 ～ 9 のグループに分け、左から順に並べる。

180、410、282、315、645、525

② 数値の 10 の位の 0 ～ 9 のグループに分け、左から順に並べる。

410、315、525、645、180、282

処理結果は②の並びになる。求める答えはウとなる。

例題演習

探索方法とその実行時間のオーダの正しい組合せはどれか。ここで、探索するデータ数を n とし、ハッシュ値が衝突する（同じ値になる）確率は無視できるほど小さいものとする。また、実行時間のオーダが n^2 であるとは、 n 個のデータを処理する時間が $c n^2$ （ c は定数）で抑えられることをいう。

	2 分探索	線形探索	ハッシュ探索
ア	$\log_2 n$	n	1
イ	$n \log_2 n$	n^2	1
ウ	n^2	1	n
エ	$n \log_2 n$	n	$\log_2 n$

解答解説

探索法の計算量に関する問題である。

次の表は、データ数に対する線形探索、二分探索、ハッシュ法の計算量を示したものである。

データ数	最大比較回数		
	線形探索	二分探索	ハッシュ法
1000	1000	10	1
10000	10000	14	1
10^8	10^8	27	1
N	N	$\log_2 N + 1$	1

2分探索のオーダは $\log_2 n$ 、線形探索のオーダは n 、ハッシュ探索のオーダは1となり、求める答えはアとなる。

例題演習

クイックソートに必要な考え方で、手続き中、自分自身を呼び出すのはどれか。

ア リューザブル

イ リエントラント

ウ リロケータブル

エ リカーシブ

解答解説

クイックソートとリカーシブの関係に関する問題である。

リカーシブは、プログラムの中から自分自身を呼び出すことを言う。自分自身を定義するのに自分自身よりも1次低い集合を用いる。その部分集合はより低次の部分定義を用いて定義することを繰り返して表現する。

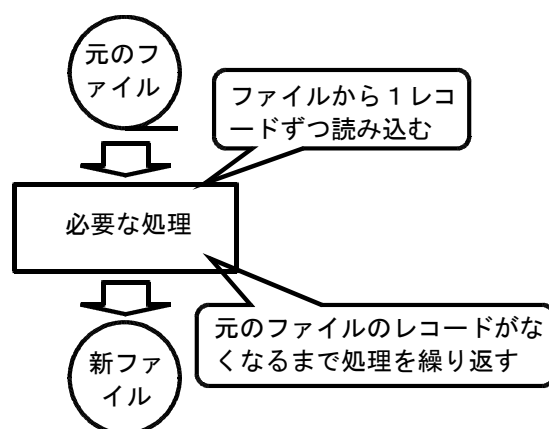
クイックソートは、適当にサンプリングして得られたデータNに着目し、N以下のデータ系列とN以上のデータ系列に分割する。このデータNを軸(ピボット)という。データNを軸として、その軸の値より小さい要素は軸より前方へ、大きな要素は軸より後方へ振り分ける。適当な長さの系列になるまで分割を繰り返し、それぞれの系列をソートすれば1つのソート済み系列となる。

クイックソートは、分割の繰返しに再帰呼び出し(リカーシブ)を利用した高速の整列法である。求める答えはエとなる。

① 単一ファイル処理

① 単一ファイル処理とは

複数のレコードが格納されているファイルから、1レコードずつ読み込み、媒体変換やレコードの検査、グループ集計処理などを行って新しいファイルを作成したり、一定の様式の帳票を出力する処理である。ファイル処理を実行する前に、ファイル内のレコードの整列を行う。グループ集計処理にはコントロールブレイク処理が利用される。



② アルゴリズムの構成

単一ファイル処理のアルゴリズムは、次の3つの部分で構成される。

㊦ 前処理

㊧ 本体処理

本体処理では、複数のレコードに対して同じ処理を繰り返す。

㊨ 後処理

③ 前処理で実行される内容

- ㊦ 入力データの記憶領域、演算結果の記憶領域、出力データの記憶領域などの必要な記憶領域の確保
- ㊧ 記憶領域の初期化
- ㊨ 関連するファイルのオープン
- ㊩ 最初のデータの入力、最初の表題行、見出し行の印刷
- ㊪ キーの記憶域への移動
- ㊫ 最初のデータの演算処理、演算結果の記憶領域への出力

㉔ 本体処理で実行される内容

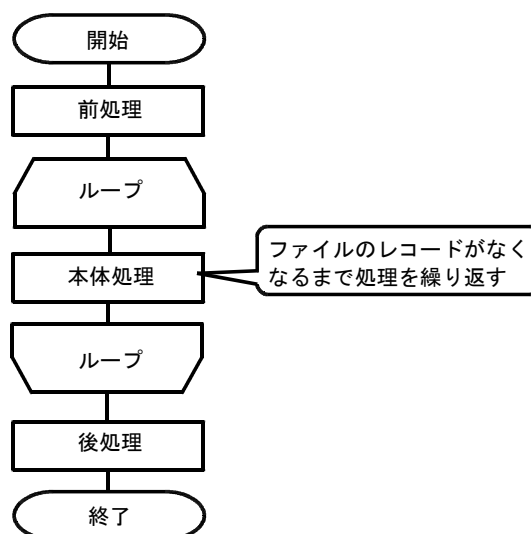
- ㊦ 入力レコードの有無の判断
- ㊧ 仕事の処理内容の分類
- ㊨ キーの比較、キーの記憶域への移動
- ㊩ 必要な演算処理、演算結果の記憶領域への出力、報告書やファイルへの出力
- ㊪ データ項目の照合、更新、削除、追加処理
- ㊫ レコードの整列、併合処理
- ㊬ 改ページ制御、空白行の制御、表題・見出し行の印刷
- ㊭ 処理件数、処理行のカウント
- ㊮ データの初期化、次のデータの入力

㉕ 本体処理の特徴

- ㊦ 一定の処理をレコードがなくなるまで繰り返す。
- ㊧ 幾つかの機能を分類したり、組み合わせたりしながら、繰り返し順次処理を進める。
- ㊨ 繰り返し処理が中子になったり、条件に従って幾つかの処理が選択的に行われる。
- ㊩ グループ集計では特定のキーが利用される。

㉖ 後処理で実行される内容

- ㊦ 必要な演算処理
- ㊧ 処理結果の出力
- ㊨ 関連ファイルのクローズ

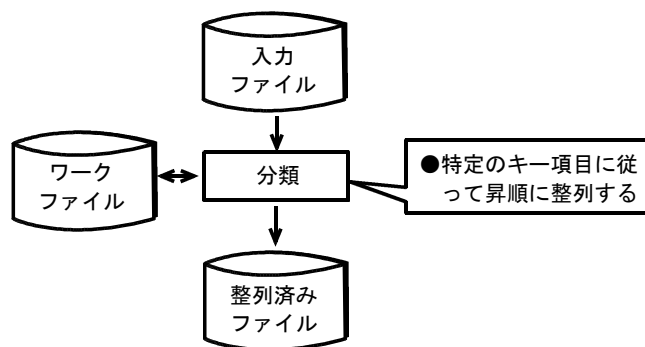


② 単一ファイル処理の基本パターン

① 入力媒体変換

入力媒体変換は、処理プロセスに最適な媒体に変換する処理で、入力媒体上のファイルを他の記録媒体に記録し直す。

② 分類処理(整列)



ファイルの中のレコードの順番を特定の項目の値や文字列の順番に従って並び換える処理である。特定の項目を整列キーといい、整列キーの小さいレコードから順に並び換えることを昇順に整列といい、その逆を降順に整列という。第一次キー、第二次キー、…、のように複数のキーを利用して整列することもある。処理以前の整列結果を利用して、最小限の整列キーを指定するだけで整列の目的が達せられることもある

③ 出力媒体変換

出力媒体変換は記録媒体上のマスタファイルから、利用者向けの媒体に見やすい体裁に出力する。対象になるファイルのレコードを1件ずつ印刷出力する明細行印刷などがある。

④ 入力レコード検査

入力レコード検査はプログラムによるデータのチェック法が使用される。数字チェック、形式チェック、限界チェック、論理チェック等の各種チェックの方法がある。

⑤ 生成処理

1つ以上のファイルを読み込んで、レコードの変形や加工を行う。

⑥ 複写処理

入力ファイルのレコードを出力ファイルに書き写す処理を行う。

⑦ 抽出処理

必要な情報を検索して抜き出す処理を行う。

③ グループ集計とコントロールブレーク処理

④ グループ制御

㊦ コントロールブレーク

グループ処理は、整列済みの入力レコードのキー項目を順次比較し、レコードのグループを見分けたり、グループ内の集計を行うことである。整列対象のキー項目をコントロール項目、同じキー項目を持つレコード群をコントロールグループ、グループの変わり目をコントロールブレークという。

① 多重グループ制御

制御対象が多重の階層をなしている場合を多重グループ制御という。多重グループ制御では複数のキーが利用され、それらのキーが整列の場合、第1次キー、第2次キー、第3次キー、…、となる。第1次キーは第2次キーよりも上位のレベルにあり、第2次キーは第3次キーよりも上位のレベルにあり、以下同様なレベル関係にある。

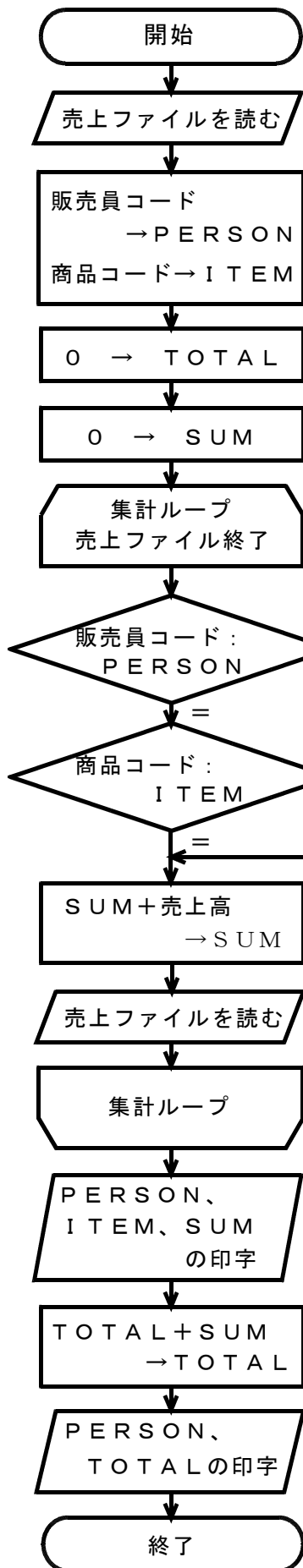
㊦ 多重レベルのコントロールブレーク処理

多重レベルのコントロールブレーク処理を行う場合、上位のレベルのキーの変化から判定し、上位のキーが不変の場合、下位のキーの判定に移る。上位のレベルのキーが変化すると、下位のレベルのキーは自動的に変化する。コントロールブレーク処理は、終わりの処理は下位のレベルから上位のレベルの順に処理し、初めの処理は上位のレベルから下位のレベルの順に処理を進める。

⑥ グループ集計処理の具体例

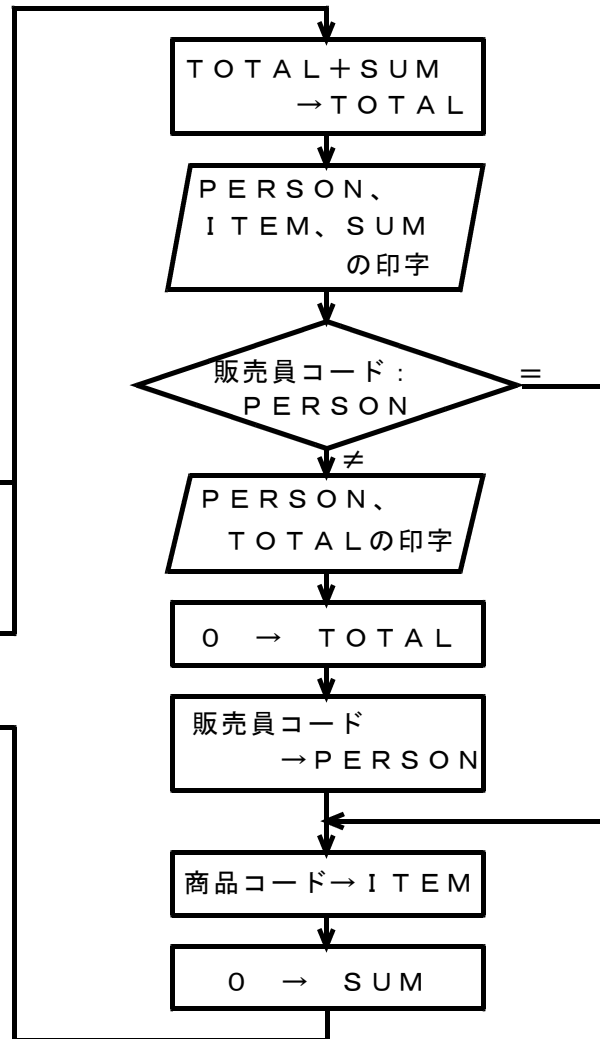
販売コード、商品コード、売上高からなる売上げファイルのレコードを読み込み、販売員別商品別売上高合計、販売員別売上高合計を印字する流れ図である。流れ図の特徴は次の通りである。

- ㊦ 第1次キーが販売員コード、第2次キーが商品コードでグループ集計処理が行われる。
- ① PERSON、ITEMは、コントロールブレーク処理の比較基準になる販売員コード、商品コードを格納する記憶領域である。
- ㊦ コントロールブレーク処理は、第1レベルの販売員コードの変化を判定し、販売員コードが不変の場合に第2レベルの商品コードの変化を判定する多重レベルのコントロールブレーク処理になっている。
- ① 販売員コードが同じ場合、商品コード別集計を行う。
- ㊦ 商品コード別集計は、売り上げファイルのレコードを読み込み、SUM+売上高→SUMを使用して集計する
- ㊦ 商品コード、販売員コードが変化すると、集計結果を印字する。
- ㊦ 売り上げファイルの最後まで集計が完了すると結果を出力する。



売上ファイルのレコード形式

販売員コード	商品コード	売上高
--------	-------	-----



課題

この売上ファイルを読み、
販売員別、商品別の売上高合計
販売員別の売上高合計
を印字する。

同一販売員コードについて同一商品コード
のレコードが複数あるものとする。

例題演習

次のコントロールブレイク処理に関する記述のうち正しいものはどれか。

- ア 入力ファイルから直接アクセスで、レコードを読み込む。
- イ ブレイクキーやレコード数が増えると、グループ集計のための記憶領域が増加する。
- ウ ブレイクの判定は上位レベルよりも下位レベルのキーから行う。
- エ コントロールブレイク処理を行うにはブレイクキーの昇順に整列させる。

解答解説

コントロールブレイク処理に関する問題である。

アの入力ファイルへのアクセスは、ブレイクキーによる昇順に整列したレコードを順次読み込むため、順次アクセスで直接アクセスの必要はない。

イのグループ集計のための記憶領域はブレイクキーの数が増加しても増加しない。キー退避領域、集計領域があれば十分である。

ウのブレイク判定の順序は、上位レベルから下位レベルの順に行う。

エのコントロールブレイク処理を行うためには、レコードの整列が必要であり、通常は昇順にブレイクキーによって整列する。求める答えはエとなる。

例題演習

マスタファイルとトランザクションファイルを照合して、トランザクションファイルの情報でマスタファイルの変動項目の更新を行う処理はどれか。

- | | |
|----------|----------|
| ア マージ | イ マッチング |
| ウ アップデート | エ メンテナンス |

解答解説

更新に関する用語を問う問題である。

アのマージは、同じファイル形式の複数のファイルを一つに統合することである。

イのマッチングは、2つ以上のデータが等しいかどうか検査することである。

ウのアップデートは、ファイル、プログラム、データベースなどに修正、追加、削除を行うことである。

エのメンテナンスは、ファイルの追加、更新や再編成である。

トランザクションファイルを利用して、マスタファイルの変動項目を更新する処理であるからアップデートであり、求める答えはウとなる。

① 複数ファイル処理

① 複数ファイル処理とは

入力ファイルが二つ以上の時、各ファイルのレコード間の論理関係を確認しながら処理を進める方法である。突合せ、併合、更新、維持などの処理がある。

② 突合せ(マッチング)

基本となるファイルのレコードの中の特定のデータ項目を取り出し、別のファイルのレコードの内容と合体させて、新しいレコードとする処理である。

③ 併合(マージ)

キー項目の大小関係に基づいて、複数個のファイルの中のレコードを昇順、または降順に組み合わせて、一つのファイルにまとめる処理である。

④ 更新(アップデート)

マスタファイルのレコードの変動項目を、トランザクションファイルのレコードの項目で更新する処理である。

⑤ 維持(メンテナンス)

マスタファイルに新規レコードを追加したり、不要になったレコードを削除したり、レコードの固定的な項目の内容を変更したり、訂正したりして、マスタファイルを最新状態に維持する処理である。

⑥ マスタファイル

内容があまり変わらないファイルで、社員マスタファイル、顧客マスタファイル、商品マスタファイルなどがある。

⑦ トランザクションファイル

ファイルの内容が取引やレコードの発生のつど変化するファイルであり、残業時間ファイル、売上ファイルなどがある。

② 複数ファイル処理の基本

① 2つのファイルの整列条件

複数ファイル処理の対象となるファイルは、両ファイルとも特定の同一項目で整列していなければならない。特定のデータ項目が複数存在する場合、第1キー、第2キー、…、が一致していなければならない。

② 2つのファイルのキー項目の比較

ファイルAのキー項目(Aキー)とファイルBのキー項目(Bキー)の大小関係を比較する。Aキー=Bキーならば、どちらかのファイルを利用して、他方のファイルの処理を実行する。Aキー<Bキーならば、ファイルAに必要な処理を実行して、ファイルAの次のレコードを読み込む。Aキー>Bキーならば、ファイルBに必要な処理を実行して、ファイルBの次のレコードを読み込む。AキーとBキーが一致しない場合は、基本的にはキー値の小さいファイルを対象に処理を行う。処理の内容はファイル処理の種類、発生条件に応じて異なる。

③ 処理の終了条件

2つのファイルの一方の処理が終了しても、他方のファイルの残りの処理を実行しなければならない。1つのファイルの処理が終了すると、処理の終了したファイルの対象になっているキー項目に最大のキー値を設定する。2つのファイルの対象のキー項目が共に最大値になったとき、2つのファイル処理を終了する。

③ ファイルの併合

① 併合

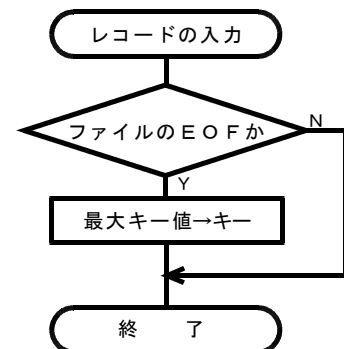
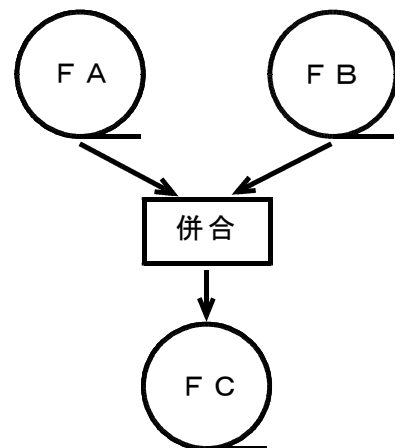
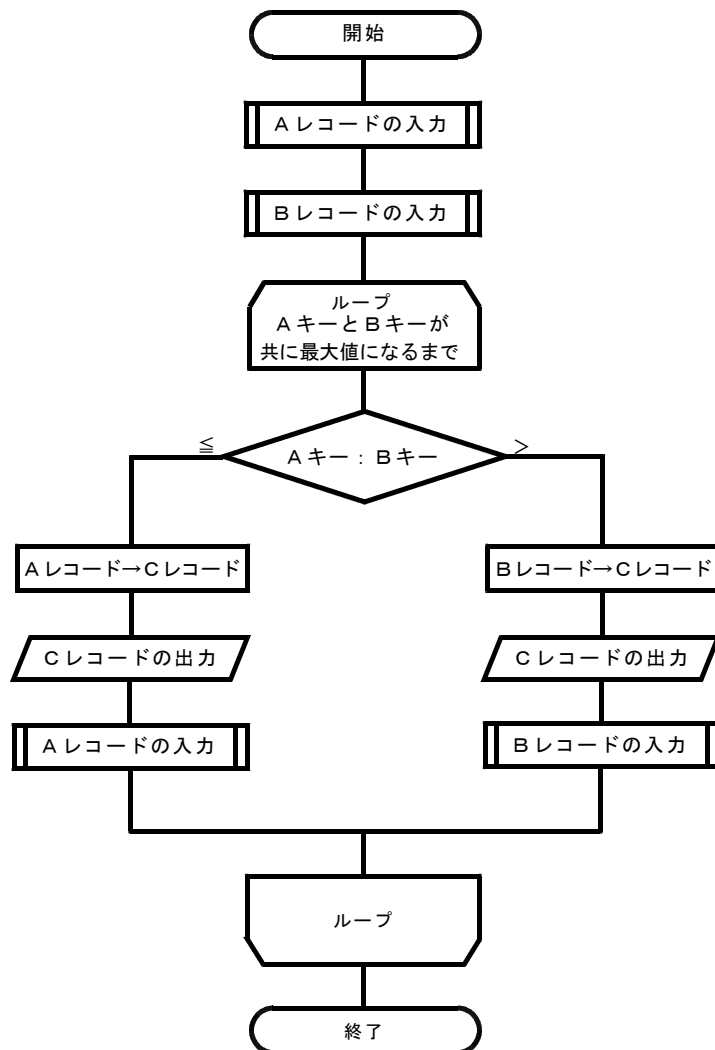
特定のキー項目の大小関係に基づいて、複数個のファイル内のレコードを昇順または降順に組み合わせて、1つのファイルにまとめることである。複数のファイルは同一キー項目は併合前に昇順に整列している。併合前後のファイルのレコード様式は共に同じ様式である。

② A、Bファイルを併合し、Cファイルを作成する手順

- ㊦ A、Bファイルから1レコードずつ読み込む。
- ㊧ 読み込んだ2つのレコードのキー項目の大小関係を比較し、キー項目の小さい方のレコードをCファイルのレコードとして出力する。
- ㊨ Aファイル(Bファイル)から読み込まれたレコードをCファイルに出力した場合は、次に読み込むレコードはAファイル(Bファイル)から読み込み、前にBファイル(Aファイル)から

読み込んだレコードとキーの値の大小関係を比較する。

- ㊦ A、Bファイルのキー項目の値が等しい場合は、あらかじめ決められた規則に従って処理を行う。
- ㊧ A、Bのファイルのレコードがなくなるまで、キー項目の比較を繰り返す。
- ㊨ ファイルのレコードが終了すると、そのファイルのキー項目に最大値を設定する。
- ㊩ A、Bのファイルが共に最大値になると、併合処理を終了する。

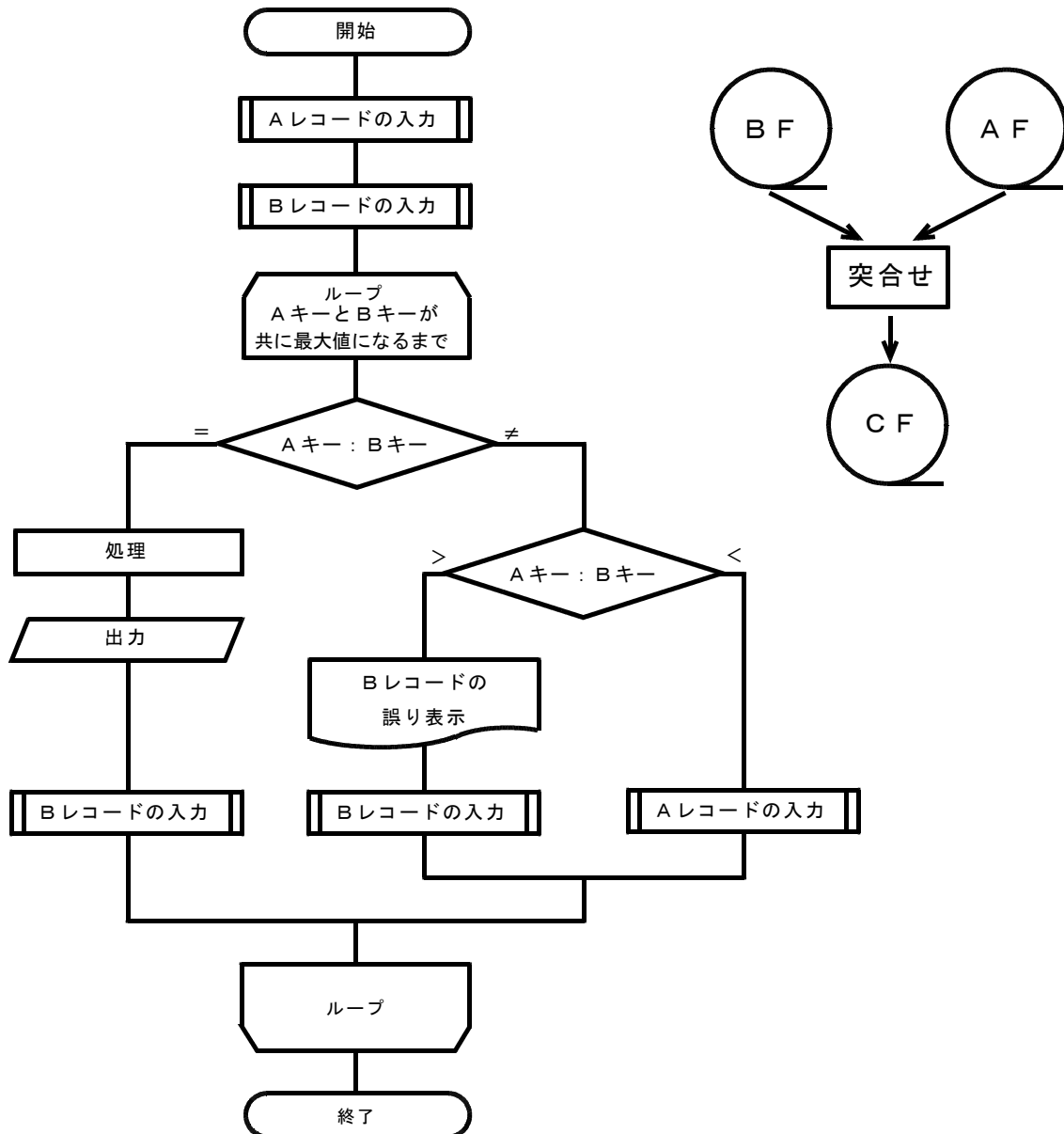


④ ファイルの突合せ

① 突合せ

突合せは、ファイルAとファイルBの特定のキー項目の一致するレコードの特定のデータ項目の照合や参照を行うことである。ファイルA、ファイルB共に同一キー項目で昇順に整列している。

② ファイルBに同一キーのレコードが複数個ある場合の突合せ手順



⑦ Aレコード、Bレコード共に同じ項目のAキー、Bキーで昇順に整列する。

① ファイルA、ファイルBからAレコード、Bレコードをそれぞれ1レコード入力する。ファイルA、ファイルBのレコードが無くなると、Aキー、Bキーの値を最大値に設定する。

- ㉔ AキーとBキーの値を比較する。
- ㉕ Aキー=Bキーならば、Bレコードに対応するAレコードが存在するので、Aレコードを利用してBレコードに必要な処理を行い、そのBレコードを出力し、次のBレコードを入力する。
- ㉖ Aキー>Bキーならば、Bレコードに対応するAレコードが存在しないため、そのBレコードに対してエラー処理を行い、次のBレコードを入力する。
- ㉗ Aキー<Bキーならば、Bレコードに対応するAレコードがそれ以降のファイルA内に存在する可能性があるため、Bレコードはそのままの状態、次のAレコードを入力する。
- ㉘ Aキー、Bキーが共に最大値になると処理を終了し、そうでなければ㉔に戻る。

⑤ ファイルの更新

㉑ 更新

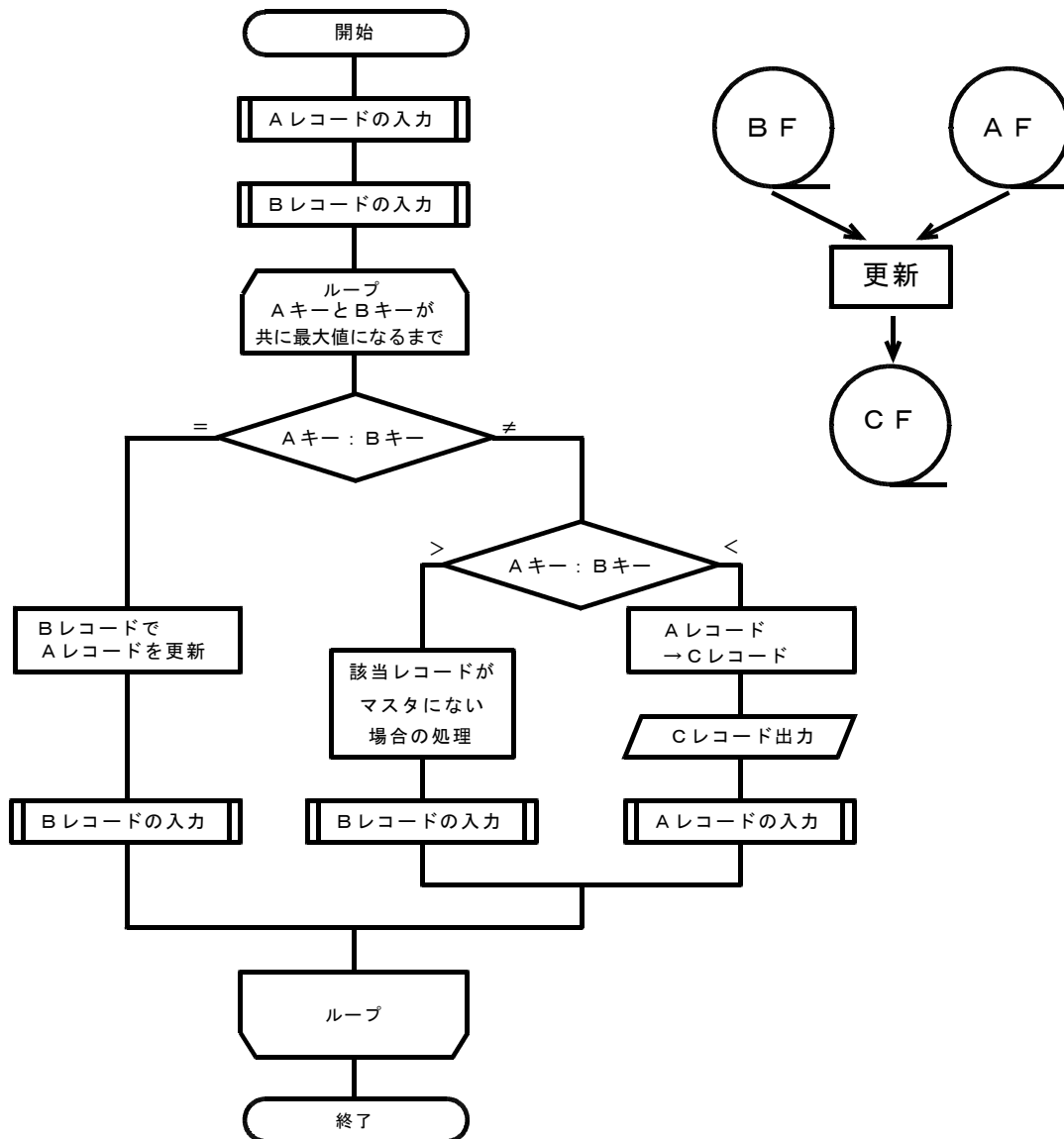
ファイルA、ファイルBの特定のAキーとBキーが一致すると、Bレコードの特定のデータ項目を利用して、Aレコードの特定のデータ項目の更新処理を行う。トランザクションファイル、マスタファイル共に同一キー項目で昇順に整列している。

㉒ ファイルBに同一キーのレコードが複数個ある場合のファイルAの更新手順

- ㉓ Aレコード、Bレコード共に同じ項目のAキー、Bキーで昇順に整列する。
- ㉔ ファイルA、ファイルBからAレコード、Bレコードをそれぞれ1レコード入力する。ファイルA、ファイルBのレコードが無くなると、Aキー、Bキーの値を最大値に設定する。
- ㉕ AキーとBキーの値を比較する。
- ㉖ Aキー=Bキーならば、Bレコードの内容を利用してAレコードの対応するデータ項目の更新処理を行い、Aレコードはそのままの状態、次のBレコードを入力する。
- ㉗ Aキー>Bキーならば、Bレコードに対応するAレコードが存在しないため、そのBレコードに対してエラー処理を行い、次のBレコードを入力する。
- ㉘ Aキー<Bキーならば、Bレコードに対応するAレコードがそれ以降のファイルA内に存

在する可能性があるため、Bレコードはそのままの状態、Aレコードを新しいファイルC
に出力し、次のAレコードを入力する。

㊦ Aキー、Bキーが共に最大値になると処理を終了し、そうでなければ㊥に戻る。



⑥ マスタファイルへのレコードの追加・削除

㊦ 追加・削除

複数ファイル処理でトランザクションファイルの内容を利用して、マスタファイルへのレコードの追加やマスタファイルからのレコードの削除を行うことができる。トランザクションファイル、マスタファイル共に同一キー項目で昇順に整列している。

⑥ ファイルBに同じBキーが複数個存在しない場合の追加・削除

- ㊦ Aレコード、Bレコード共に同じ項目のAキー、Bキーで昇順に整列する。
- ㊧ ファイルA、ファイルBからAレコード、Bレコードをそれぞれ1レコード入力する。ファイルA、ファイルBのレコードが無くなると、Aキー、Bキーの値を最大値に設定する。
- ㊨ AキーとBキーの値を比較する。
- ㊩ Bキー=Aキーならば、AキーのレコードをファイルAから削除する。削除後、ファイルBから新しいBレコードを、ファイルAから新しいAレコードを入力する。
- ㊪ Aキー>Bキーならば、Bキーに対応するAレコードがファイルA内に存在しないため、入力したAレコードはそのままの状態、Bレコードを利用して新しいAレコードを作成し、ファイルAに追加し、ファイルBから新しいBレコードを入力する。
- ㊫ Aキー<Bキーならば、Bレコードはそのままの状態、Aレコードを新しいファイルAに出力し、ファイルAから新しいAレコードを入力する。
- ㊬ Aキー、Bキーが共に最大値になると処理を終了し、そうでなければ㊨に戻る。

例題演習

16個のデータが次のような順序で順編成ファイルに格納されている。

1, 3, 5, 7, 2, 4, 6, 9, 8, 11, 13, 15, 10, 12, 14, 16

このファイルを入力として、データをマージソートで整列するのに、磁気テープ装置3台を使うことにする。入力ファイルを磁気テープ1に割り当て、出力ファイルを磁気テープ2、3に割り当てて、最初の分配を行ったとき、磁気テープ2に書かれているデータの順序として正しいものはどれか。ここで、磁気テープ2の方から先に書き始めるものとする。

- ア 1, 2, 3, 4, 5, 6, 7, 9
- イ 1, 3, 2, 4, 8, 11, 10, 12
- ウ 1, 3, 5, 7, 8, 11, 13, 15
- エ 1, 3, 5, 7, 9, 11, 13, 15, 16

解答解説

順編成ファイルを利用した併合に関する問題である。

与えられた磁気テープ1の16個のデータは昇順に整列していないため、磁気テープ2および3を使用して昇順に整列する。磁気テープ2に16個の最初のデータから、数値が増加している

データを磁気テープ2に順次出力し、数値が減少すると、それ以降の出力先を磁気テープ3に切り替える。数値が増加しているデータが連続している限り磁気テープ3に順次出力する。数値が減少すると、それ以降の出力先を磁気テープ2に切り替える。数値が増加しているデータが連続している限り磁気テープ2に、順次、出力する。

上記の手順を実行すると次のようになる。

- ① 1、3、5、7を磁気テープ2に出力する。
- ② 2、4、6、9を磁気テープ3に出力する。
- ③ 8、11、13、15を磁気テープ2に出力する。
- ④ 10、12、14、16を磁気テープ3に出力する。
- ⑤ 磁気テープ2のデータの内容は、1、3、5、7、8、11、13、15となる。

求める答えはウとなる。

例題演習

特定のキー項目の大小関係に基づいて、複数個のファイル内のレコードを昇順または降順に組み合わせて1つのファイルにまとめて処理するのはどれか。

ア 併合

イ 突合せ

ウ 維持・保守

エ 更新

解答解説

複数ファイル処理の併合に関する問題である。

アの併合は、レコードがキー項目の昇順に並んだ複数のファイルを組み合わせて、同じ並びの1つのファイルにすることである。求める答えはアである。

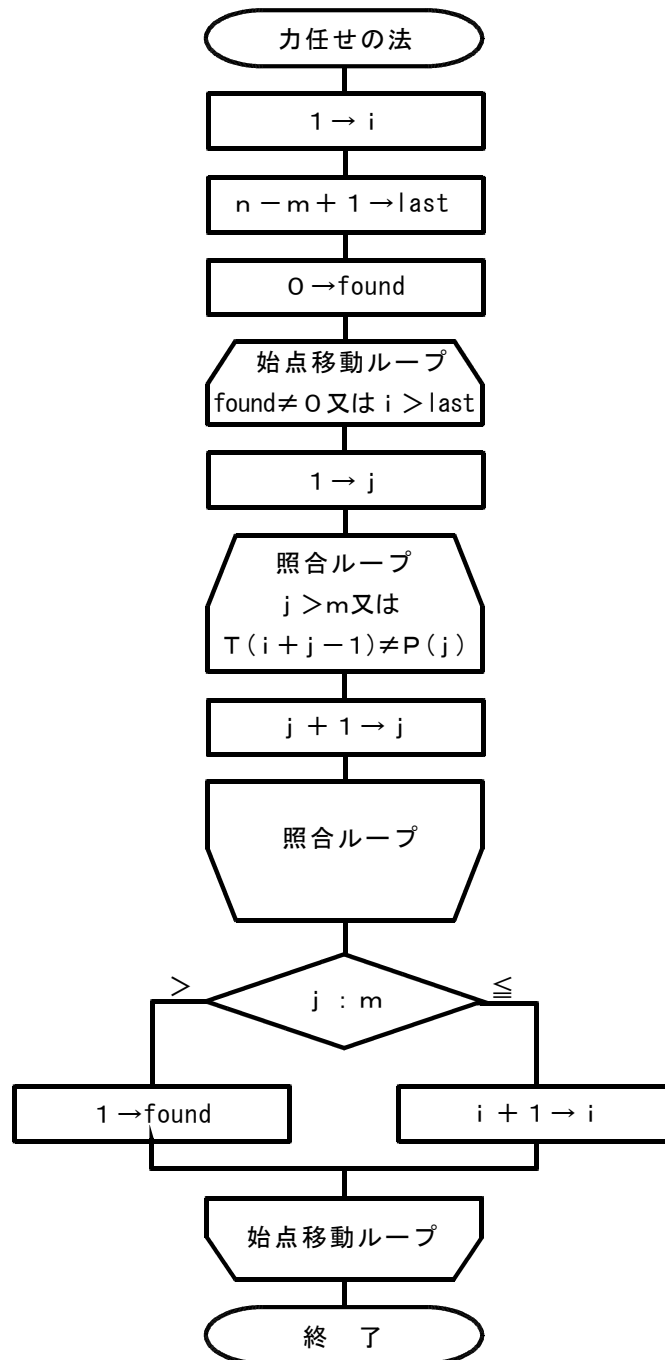
イの突合せは、同じキー項目で昇順に整列した2つのファイルを照合して、レコードが等しいかどうかを判定することである。

ウの維持・保守は、ファイルの追加、更新や再編成である。

エの更新はファイルやデータベースのレコードを新しいレコードに置き換えることである。

① 力任せの探索法

② 力任せの探索法(文字列の照合)とは



テキスト文字列の中にパターン文字列と同じ長さの部分文字列をつくり、パターン文字列の対応する文字と、順次比較し、全文字が一致すれば探索成功であり、比較途中で不一致になると、配列の次の要素に移動して、新しい部分文字列を作りパターン文字列と比較する。テキスト文字列のすべての要素と比較して不一致ならば探索不成功となる。

⑥ 力任せの探索法の手順

- ㉞ 一つのポインタ i をテキスト上に、もう一つのポインタ j をパターン上に置き、 i および j を 1 で初期化する。
- ㉟ テキスト上にポインタ i を先頭に長さ m の部分文字列を作成し、部分文字列の要素の添字を $i + j - 1$ で表す。
- ㊱ 添字 j を操作して、テキスト上の部分文字列とパターンの対応する要素を比較する。
- ㊲ 対応する要素の内容が一致するとき、 j のインクリメントを行い、パターンの長さに等しい一致が生じると照合が得られる。
- ㊳ 対応する要素の内容が比較途中で不一致になるかまたは照合が得られた場合、パターンの添字 j をリセットし、テキスト上のポインタ i をインクリメントし、㉟に戻る。
- ㊴ i がテキスト上の $n - m + 1$ の要素を超えると文字列の探索を終了する。

⑦ 流れ図の内容

- ㉞ テキスト文字列の i 番目の文字を先頭に長さ m の部分文字列を考え、テキストの部分文字列の要素を添字 $i + j - 1$ で表す。
- ㉟ i を 1 で、 $last$ を $n - m + 1$ で、 $found$ を 0 で初期化する。
- ㊱ j を 1 で初期化して照合ループに入る
- ㊲ $T(i + j - 1) = P(j)$ ならば、 $j > m$ になるまで、 $j + 1 \rightarrow j$ の操作を繰り返す。
- ㊳ $T(i + j - 1) \neq P(j)$ ならば、照合ループから脱出し、 $i + 1 \rightarrow i$ を実行し、 i を先頭とする新しい部分文字列を考え、㉟に戻る。
- ㊴ $T(i + j - 1) = P(j)$ を繰り返し、 $j > m$ となると照合成立となり、照合ループから脱出し、 $1 \rightarrow found$ となり、始点移動ループから脱出する。
- ㊵ 始点移動が進み、 $i > n - m + 1$ となると、照合不成功で探索を終了する。

⑧ 力任せの探索法の比較回数

テキストの長さを n 、パターンの長さを m とすると、力任せの探索法の最悪の場合の比較回

数は $(n - m + 1) \times m$ となる。テキストの長さ n の値が、パターンの長さ m に比べて、大きい場合は nm となる。

② ボイヤ・ムーアの方法

① ボイヤ・ムーアの文字列探索法

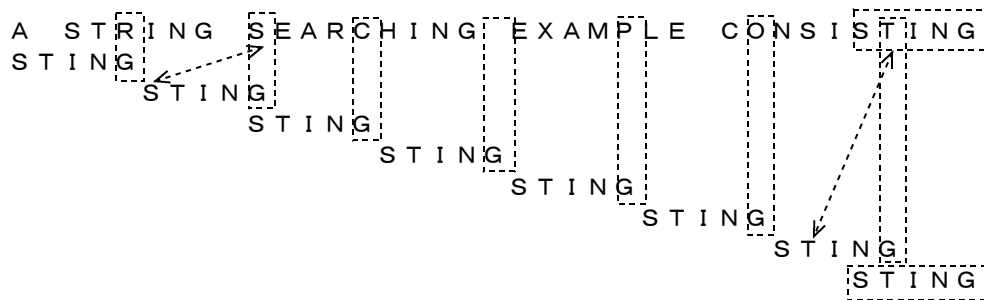
パターンをテキストと照合させる時に右から左へポインタを動かす走査を取り入れることによってかなり速い文字列探索法を開発することができる。パターンだけでなく不一致を生じたテキスト文字も考慮してスキップの文字数を決定する。スキップの文字数は、テキストの各要素 $T(i)$ に対して、移動距離として、配列 $d(T(i))$ に設定する。

② パターン長 5 (STING) の場合の具体例

- ㊦ まず、最初にテキストの左端の 5 文字とパターンの 5 文字を比較する。
- ㊧ パターンの右端の文字とそれに対応するテキスト上の文字を比較し、一致すると、㊦の操作に移る。
- ㊨ 一致しなければ、テキストとパターンの比較はパターンの右端の文字 G に対応するテキスト上の文字 R とパターンの各文字を比較する。
- ㊩ テキスト上の文字 R がパターンの中に現れない文字であることがわかると、パターン全体を 5 文字右に移動する。
- ㊪ 次に、テキストの 10 番目の文字 S とパターンの文字を比較する。文字 S とパターンの右端の文字が不一致でパターンの先頭の文字 S が一致する。
- ㊫ そこで、テキストの文字 S とパターンの S が一致するまでパターンの文字列全体を右に 4 文字移動させる。
- ㊬ 次に、テキスト上文字 C とパターン文字列を比較する。テキスト上文字 C がパターンの中に存在しないことが分かる。そこで、パターン文字列を右に 5 文字分移動し、同じことを繰り返す。
- ㊭ 同様の移動を 3 回行くと、CONSISTING の T に到着する。
- ㊮ このテキスト上の文字 T がパターンの右端の文字と不一致であるが、パターンの中に存在するため、パターン文字列の T がテキスト上の文字 T と一致するまでパターンを右に移動さ

せる。

- ㉓ テキスト上の文字Gがパターン上の右端の文字Gと一致する。
- ㉔ テキスト上の文字とパターン上の右端の文字が一致すると、それぞれの文字列の比較対象文字を一つ前の文字に移動し、二つの文字の比較を繰り返す。
- ㉕ テキストの文字列とパターン上の文字列が完全に一致すると、探索成功になる。
- ㉖ テキスト文字列とパターン文字列の不一致が途中で発生すると、パターン文字列の右端の文字に対応するテキスト上の文字と同じ文字がパターン上にあれば、二つの文字が一致するまで、パターン文字列を右に移動させる。
- ㉗ パターン文字列の右端の文字に対応するテキスト上の文字と同じ文字がパターン上になければ、パターン文字列の長さ分だけパターン文字列を右に移動させる。



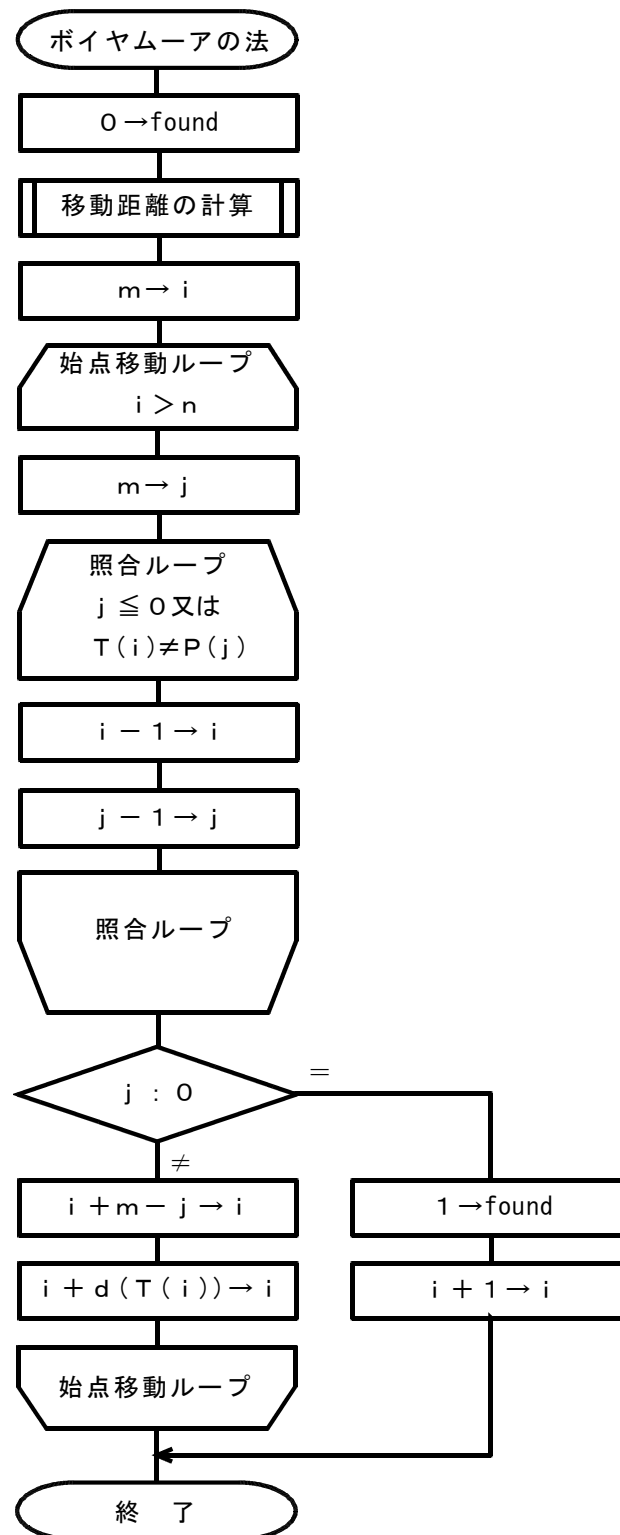
㉘ 文字列探索の要領

- ㉙ テキスト文字列の最初の比較文字は先頭からm番目とする。
- ㉚ パターン文字列の最初の比較文字は最後のm番目の文字からはじめる。
- ㉛ テキストの部分文字列の要素とパターン文字列の要素とが一致する間、 i および j のディクリメントを行い比較対象を変える。
- ㉜ $T(i) \neq P(j)$ となると、照合を中断して、照合ループから脱出する。
- ㉝ 照合ループを脱出後、テキストの文字列の位置を最初のmの位置に戻す。
- ㉞ 照合開始時のテキストの文字列の要素 $T(i)$ に対する移動距離を求める。
- ㉟ 次の照合の開始点を $i + d(T(i))$ を求める。

② 照合を繰り返し、 $j \leq 0$ となると、照合ループから脱出し、 $1 \rightarrow \text{found}$ として、更に $i + 1 \rightarrow i$ を求めて、パターン文字列と一致するテキスト上の位置を決め、処理を終了する。

③ ⑤で新しく求めた開始点を起点として、②～④の操作を繰り返す。

④ ②～③の操作を繰り返して、 $i > n$ となると、照合不成立で処理を終了する。



㉔ 移動距離の計算要領

- ㊦ テキストの部分文字列とパターン文字列を比較する場合のテキスト文字列の最初の文字がパターン文字列の最後尾にあるか、または、パターン文字列の中に存在しない場合、パターン文字列の長さ m だけ移動する。
- ㊧ 比較するテキスト文字列の最初の文字がパターン文字列の中にある場合、二つの文字の移動距離を求め、移動距離分だけ開始点を移動する。
- ㊨ 比較するテキスト文字列の最初の文字がパターン文字列の中に複数個ある場合、最後尾の文字までの移動距離を使用する。

㉕ ボイヤ・ムーアの比較回数

ボイヤ・ムーアの文字列探索における文字の比較回数の上限は $M + N$ に比例する。テキスト文字列の文字の種類がパターンの長さに比べて十分多ければ、比較回数は N / M 回である。平均的な比較回数は N / M となる。パターンの中に現れる文字は、アルファベットの一部にすぎないことが多い。このような場合、比較が行われるごとに M 文字スキップされるようになる。このアルゴリズムは最悪の場合は線形となる。

③ 文字列の圧縮

㉖ 文字列の圧縮

同一の文字が3文字以上連続する場合、次の3文字の様式に置き換える。

圧縮マーク＋連続する同じ文字の個数＋連続する同じ文字の内容

圧縮後の文字列を同じ配列上に設定する場合と異なる配列上に設定する場合がある。同じ配列上に設定する場合は、圧縮後の文字列の詰め合わせを実行しながら圧縮処理を行う必要がある。

㉗ 文字列圧縮の手順

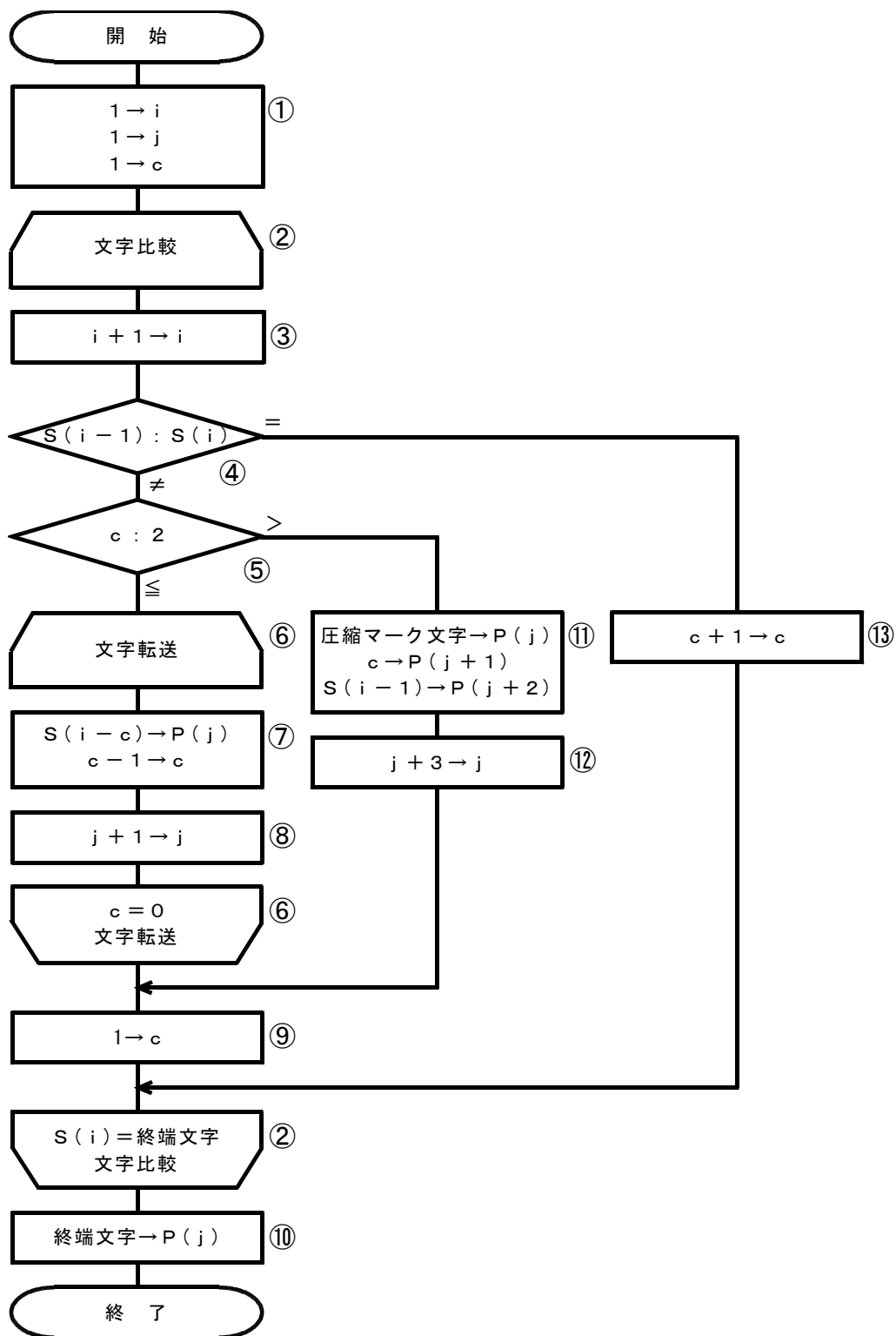
- ㊦ 文字列の先頭から隣り合う2つの配列の要素を比較して、同一文字が連続する間、連続する同じ文字の個数を求める。
- ㊧ 配列の隣り合う文字列が異なる場合に、文字の転送または文字列の圧縮を行う。
- ㊨ 連続する同じ文字数が2以下の場合は圧縮せずにそのまま転送する。
- ㊩ 連続する同じ文字数が3以上の場合、圧縮記号＋同じ文字数＋同じ文字の内容の3文字で

表す。

- ㊸ 文字列の最後になると、終端記号を付加して終了する。

㉔ 文字列圧縮の流れ図の内容

- ㊶ 配列 S の添字 i 、配列 P の添字 j 、連続する同じ文字数を表す変数 c を 1 で初期化する。
- ㊷ 文字列の比較の繰り返しに入る。終端文字が現れるまで文字列比較を繰り返す。終端文字が現れると、処理㊸に移る。
- ㊹ i をインクリメントし、着目する配列の要素を右隣に移動させる。
- ㊺ 配列 S の着目する要素 i と一つ前の要素 $(i - 1)$ の文字を比較する。2 つの文字が等しい場合は処理㊻に移り、等しくない場合は処理㊼に移る。
- ㊽ 連続する同じ文字の個数 c と 2 を比較して、 $c > 2$ の場合は処理㊾に移り、 $c \leq 2$ の場合は処理㊿に移る。
- ㊿ 文字転送操作の繰り返しに入る。 c が 2 以下の場合、同じ文字数 c を配列 S から配列 P に移動させる。従って、 c が 0 になるまで転送操作㊶～㊹を繰り返す。
- ㊶ 配列 S の要素 $(i - c)$ の文字を配列 P の要素 j に転送し、連続する同じ文字数 c をディクリメントする。
- ㊷ 配列 P の要素の添字 j をインクリメントし、次の転送位置に移動する。
- ㊸ $c \leq 2$ の場合の全ての文字の転送、 $c > 2$ の場合の文字列の圧縮の操作が完了すると、配列 S の連続する同じ文字の個数 c を 1 で初期化する。
- ㊹ 配列 P の要素の添字 j に終端文字を格納する。
- ㊺ 配列 S の文字列の同じ文字が 3 個以上連続する場合、3 個の文字列に圧縮し、配列 P に格納する。P (j) に圧縮マーク文字、P ($j + 1$) に同じ文字数の c の値、P ($j + 2$) に同じ文字の内容 S ($i - 1$) の文字を格納する。
- ㊻ 配列 P の要素の次の挿入位置 $j + 3$ を求める。
- ㊼ 連続する同じ文字の個数 c をインクリメントする。



④ 文字列の置換

④ a 文字列の置換

テキスト文字列においてパターン文字列に一致する文字列を探索し、一致文字列を置換文字

列に置き換える処理である。文字列の探索は、テキスト文字列上の部分文字列とパターン文字列の対応する要素を比較して、パターン文字列の大きさと同じ連続個数一致すれば探索成功となる。処理結果を同じ配列上に設定する場合は、パターン文字列の大きさと置換文字列の大小関係によってテキスト文字列の挿入・削除の処理が必要になる。パターン文字列の大きさが置換文字列の大きさよりも大きい場合は詰め合わせが必要となり、削除の考え方を利用する。置換文字列の大きさがパターン文字列の大きさよりも大きい場合は空白が必要となり、挿入の考え方を利用する。

⑥ 文字列置換の流れ図

大きさ N の配列 A のテキスト文字列中に、長さ M の配列 B のパターン文字列が存在すると、その部分を長さ L の配列 C の文字列で置換する。配列 A の定義領域は無限として、置換は配列 A 上で行う。また、 M と L の大小関係は不定である。

流れ図は次の主要部分で構成される。

- ㉖ 配列 A のテキスト文字列中の配列 B のパターンを探索する部分
- ㉗ 配列 A のテキスト文字列中に配列 C の文字列を挿入するための準備の部分
- ㉘ 配列 A のテキスト文字列中に配列 C の文字列を挿入する部分

⑦ 文字列置換の流れ図の内容

㉖ ループ A の機能

配列 A の部分文字列の先頭位置を $1 \sim N - M + 1$ まで走査する。終了条件は、 $I > N - M + 1$ となる。

㉗ ループ B の機能

配列 A の部分文字列と配列 B のパターン文字列を比較する。配列 A の部分文字列は添字 I で始まる文字列で、文字列の操作は配列 B と同じ J を使用する。配列 A の部分文字列の要素は $(I + J - 1)$ で表すことができる。配列 $A(I + J - 1)$ 、配列 $B(J)$ とともに、 $J = 1$ で初期化し、 J のインクリメントを使用して要素間を移動する。対応する両者の文字列が不一致になると、ループ B から脱出するための SW を使用する。最初は、 $SW = 0$ で初期化し、不一致になると $SW = 1$ となる。配列 $A(I + J - 1)$ 、配列 $B(J)$ の要素の比較が、 $J > M$ になると、二つの文字列が一致する。

㉘ ループ C の機能

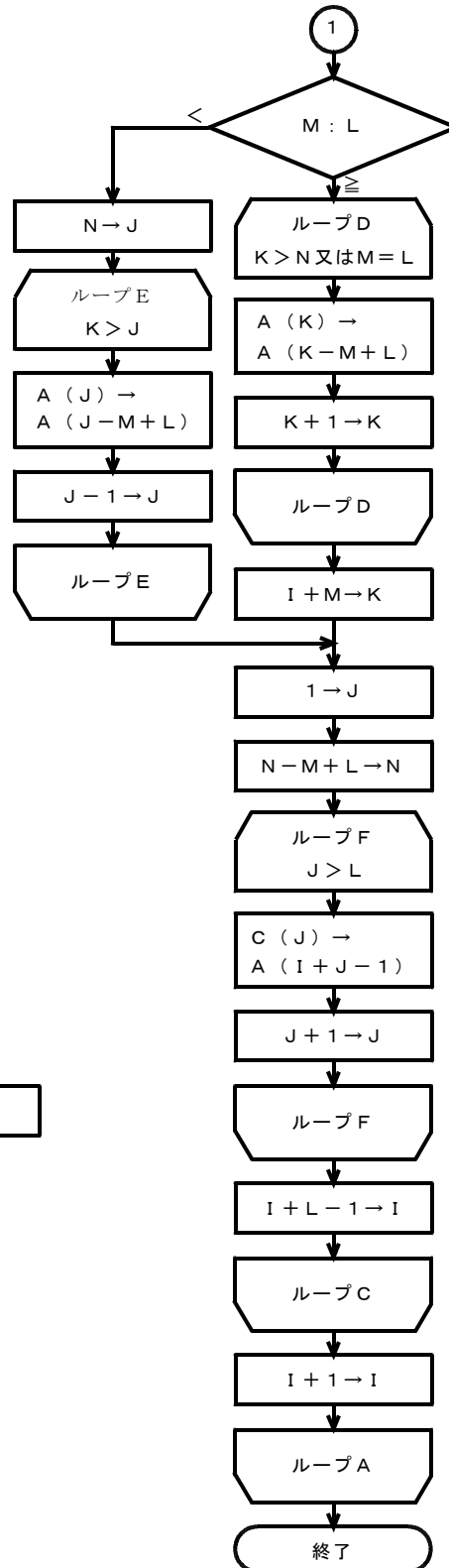
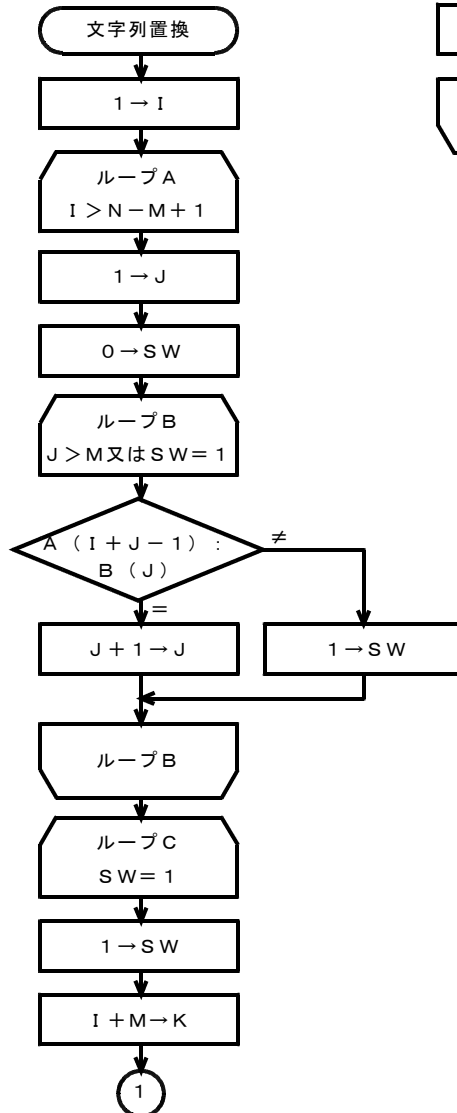
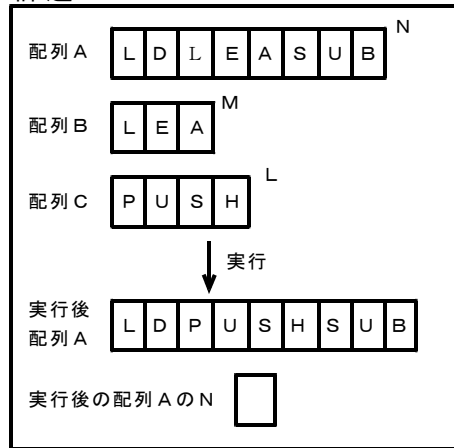
ループ B の脱出条件が $SW = 1$ の場合、グループ C を実行しない。スイッチングの役割を果たす

㉙ ループ D の機能

$M > L$ の場合、配列の削除の考え方を利用して置換領域の詰め合わせを行う。配列 A の部

分文字列の先頭位置 I から後方の $I + M - 1$ の範囲は置換対象となるため、詰め合わせの最初の移動対象の位置は $I + M \rightarrow K$ となる。この K を新たな添字として利用し詰め合わせの処理を行う。配列 A の添字 $I + M$ から N までの範囲を $M - L$ 前に詰める。 $A(K) \rightarrow A(K - M + L)$ を実行する。 $M = L$ の場合は詰め合わせの処理が不必要である。

課題



㊦ ループEの機能

$M < L$ の場合、配列の挿入の考え方を利用して置換領域を後方に広げる。移動対象の添字 J を N で初期化して、配列 A の後方から $I + M$ の範囲を $L - M$ 後方にずらせる操作を行う。 $A(J) \rightarrow A(J - M + L)$ 、 J のディクリメント $J - 1 \rightarrow J$ を実行する。終了条件は、㊥で定義した K を使用して $J < K$ となる。

㊧ ループFの機能

配列 C の文字列を配列 A の中に置換する。 $C(J) \rightarrow A(I + J - 1)$ を実行する。

例題演習

文字列 $P = p_1 p_2 \cdots p_m$ の中に文字列 $T = t_1 t_2 \cdots t_n$ が部分列として含まれているか否かを判定する。 P の部分列 $P_1 P_2 \cdots P_n$ 、 $P_2 P_3 \cdots P_{n+1}$ 、 $\cdots$ に対して順に T と一致するかどうかを調べると、文字の照合回数は、どのオーダーになるか。ここで $n \ll m$ とする。

ア m

イ mn

ウ mn^2

エ $n \log m$

解答解説

力任せの文字列処理の文字照合回数を求める問題である。

力任せの文字列処理は、テキスト文字列の各要素を先頭にパターン文字列と同じ長さの部分文字列を作成し、部分文字列とパターン文字列の対応する要素を比較し照合する。従って、パターン文字列の長さの文字列がテキスト文字列の長さの個数だけできることになる。

テキストの長さを m 、パターンの長さを n とすると、テキストの中にできるパターン文字列に対応するブロックの個数は最悪の場合は $(m - n + 1)$ 個となる。各ブロックとパターン文字列との比較回数は最悪の場合 n 回であるから、力任せの探索法の最悪の場合の比較回数は $(m - n + 1) \times n$ となり、 m の値が大きい場合は mn となる。求める答えはイとなる。

例題演習

相異なる n 個のデータが昇順に整列された表がある。この表を 1 ブロック m 個に分割し、各ブロックの最後尾のデータだけ線形探索することによって、目的のデータの存在するブロックを探し出す。次に当該ブロック内を線形探索して目的のデータを探し出す。このときの平均探索回数はどれか。ここで、 $m < n$ とし、目的のデータは必ず表の中に存在するものとする。

ア $\frac{n}{m}$

イ $\frac{n}{2m}$

ウ $m + \frac{n}{m}$

エ $\frac{m}{2} + \frac{n}{2m}$

解答解説

線形探索の計算量の問題である。

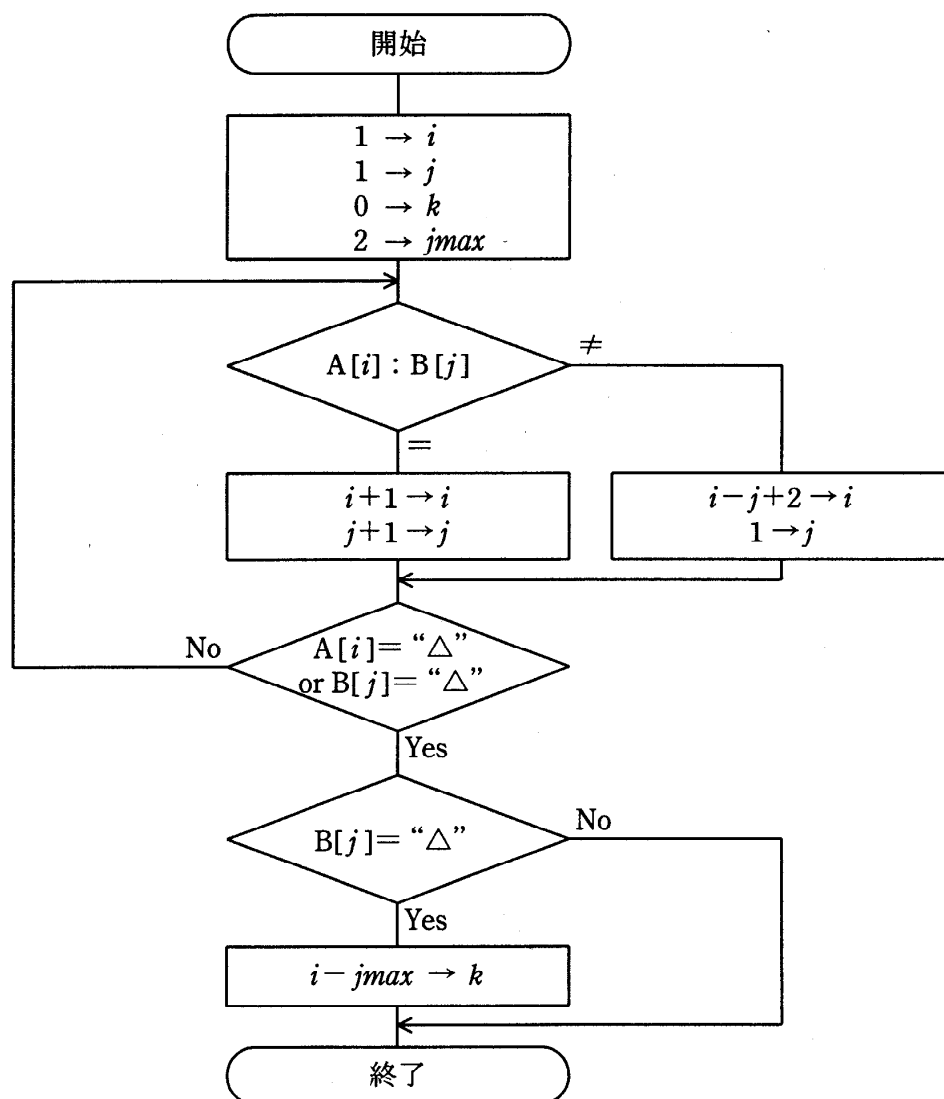
n 個のデータを 1 ブロック m 個に分割すると、ブロック数は最大 $n / m + 1$ 個となる。探索

するデータがどのブロックに属するかを調べるために、各ブロックの最後尾の値と比較する線形探索を用いると平均比較回数は $n/2m$ 回となる。各ブロックの中も線形探索を実行するため平均探索回数は $m/2$ となる。全平均比較回数は、 $n/2m + m/2$ となる。求める答えはエとなる。

この種の問題としては、探索成功確率と組み合わせた問題、2分探索と線形探索を組み合わせた場合の平均探索回数を求める問題などがある。

例題演習

文字列Aが“a a b a b x Δ”，文字列Bが“a b Δ”であるとき、流れ図の終了時点のkは幾らか。ここで、文字列の先頭の文字を1番目と数えるものとし、A[i]はAのi番目の文字を、B[j]はBのj番目の文字を、“Δ”は終端を示す文字を表す。



ア 0

イ 1

ウ 2

エ 4

解答解説

流れ図に関する問題である。

配列に格納された2つの文字列を照合する問題である。

照合の手順

- ① 文字列A、文字列Bの先頭から対応する要素を比較する。
- ② 2つの要素が一致すると、文字列A、文字列B共に添字のインクリメントを行い、④に移る。
- ③ 2つの要素が一致しない場合、 $i - j + 2 \rightarrow i$ で文字列Aの先頭位置を求め、文字列Bは先頭位置に戻る。
- ④ 文字列Aまたは文字列Bのいずれかがスペースになると照合を終了し、スペースでない場合、①の処理に戻る。
- ⑤ 照合が終了した場合、文字列Bのスペースで終了すると、 $i - j \max \rightarrow k$ を求める。そうでなければ何もしないで終了する。

文字列A = a a b a b x Δ、文字列B = a b Δとして照合手順を実行する。

- ① a = aで一致
 - ② a ≠ bで不一致、 $i = 2$ 、 $j = 2$ 、 $i - j + 2 \rightarrow 2$ となる。
 - ③ a = aで一致
 - ④ b = bで一致
 - ⑤ a = Δで不一致、 $i = 4$ 、 $j = 3$ 、 $i - j + 2 \rightarrow 3$ となる。
 - ⑥ b[3] = Δで照合を終了し、 $i = 4$ 、 $j \max = 2$ であるから、 $k = 4 - 2 = 2$ となる。
- 求める答えはウとなる。

例題演習

長さm、nの文字列を格納した配列X、Yがある。図はこれらの文字列をこの順に連結した文字列を配列Zに格納する処理の流れ図である。a、bに入れる処理として、正しいものはどれか。

ここで、1文字が一つの配列要素に格納されるものとする。

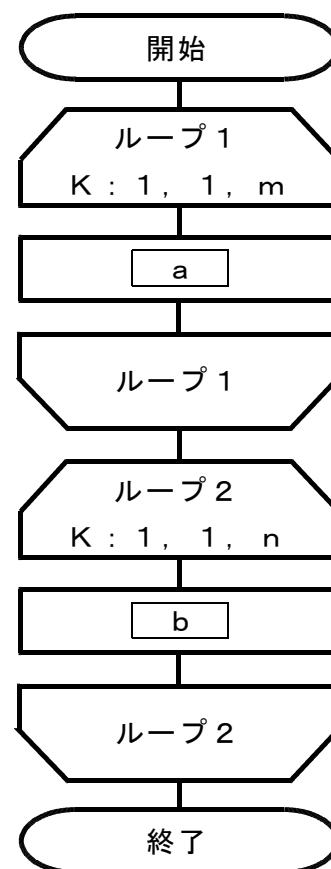
	a	b
ア	$X(k) \rightarrow Z(k)$	$Y(k) \rightarrow Z(m+k)$
イ	$X(k) \rightarrow Z(k)$	$Y(k) \rightarrow Z(n+k)$
ウ	$Y(k) \rightarrow Z(k)$	$X(k) \rightarrow Z(m+k)$
エ	$Y(k) \rightarrow Z(k)$	$X(k) \rightarrow Z(n+k)$

解答解説

2つの配列を結合する流れ図に関する問題である。

結合する手順は次の通りである。

- ① 最初に大きさmの配列X(k)を配列Zに格納する。
- ② 配列Xを格納後の配列Zの大きさはmである。
- ③ その次に大きさnの配列Y(k)を配列Zに格納する。
- ④ 配列YはZ(m+1)からZ(m+n)の要素に格納される。



⑤ 配列 X、Y を格納後の配列 Z の大きさは $m+n$ となる。

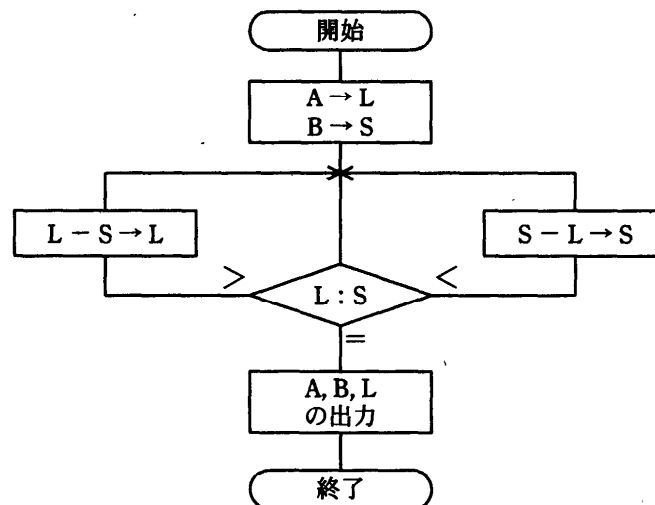
ループ 1 は、配列 X から配列 Z への文字列の格納処理である。従って、配列 X の長さは m であるから、 $X(k) \rightarrow Z(k)$ となる。

ループ 2 は、配列 Y から配列 Z の $m+1$ 番目の要素から n 個の要素を配列 Z に格納し、配列の要素番号 $m+n$ までの要素に格納する。即ち、 $Y(k) \rightarrow Z(m+k)$ となる。

a、b の組み合わせはアとなる。求める答えはアとなる。

例題演習

次の流れ図は、2 数 A、B の最大公約数を求めるユークリッドの互除法を、引き算の繰返しによって計算するものである。A が 876、B が 204 のとき、何回の比較で処理は終了するか。



ア 4

イ 9

ウ 10

エ 11

解答解説

ユークリッド互除法の流れ図に関する問題である。

- ① $L=876$ $S=204$ $L > S$ $L - S = 876 - 204 = 672 \rightarrow L$
- ② $L=672$ $S=204$ $L > S$ $L - S = 672 - 204 = 468 \rightarrow L$
- ③ $L=468$ $S=204$ $L > S$ $L - S = 468 - 204 = 264 \rightarrow L$
- ④ $L=264$ $S=204$ $L > S$ $L - S = 264 - 204 = 60 \rightarrow L$
- ⑤ $L=60$ $S=204$ $L < S$ $S - L = 204 - 60 = 144 \rightarrow S$
- ⑥ $L=60$ $S=144$ $L < S$ $S - L = 144 - 60 = 84 \rightarrow S$
- ⑦ $L=60$ $S=84$ $L < S$ $S - L = 84 - 60 = 24 \rightarrow S$
- ⑧ $L=60$ $S=24$ $L > S$ $L - S = 60 - 24 = 36 \rightarrow L$
- ⑨ $L=36$ $S=24$ $L > S$ $L - S = 36 - 24 = 12 \rightarrow L$
- ⑩ $L=12$ $S=24$ $L < S$ $S - L = 24 - 12 = 12 \rightarrow S$
- ⑪ $L=12$ $S=12$ $L = S$ $S - L = 12 - 12 = 0$

従って、11 回となり、求める答えはエとなる。

例題演習

次の方法によって、データに検査数字（チェックディジット）を付加する。データにエラーが含まれていない場合、 $N_2=7$ 、 $N_3=6$ 、 $N_4=2$ 、 $C=4$ のとき、 N_1 の値は幾らか。

元のデータ : $N_1 N_2 N_3 N_4$

検査数字 : $C = \text{mod}((N_1 \times 1 + N_2 \times 2 + N_3 \times 3 + N_4 \times 4), 10)$

ここで、 $\text{mod}(x, 10)$ の値は、 x を10で割った余り

検査数字を付加したデータ : $N_1 N_2 N_3 N_4 C$

ア 0

イ 2

ウ 4

エ 6

解答解説

チェックディジットに関する問題である。

N_1 の値を X とすると、次の式が成立する。

$$\text{mod}(X + 7 \times 2 + 6 \times 3 + 2 \times 4, 10) = 4$$

$$X + 7 \times 2 + 6 \times 3 + 2 \times 4 + (10 - 4) = X + 14 + 18 + 8 + 6 = X + 46$$

$X + 46$ が10の倍数になる X の値を求めればよい。

アの場合は46、イの場合は48、ウの場合は50、エの場合は52となる。求める答えはウとなる。

例題演習

試験の可否を判定する次の決定表から読み取れるものはどれか。ここで、試験は労務管理、経理及び英語の3科目で構成され、それぞれの満点は100とする。

ア 英語の得点が90以上の者は、

仮合格か合格になる。

イ 英語の得点が90未満の者は、

不合格になる。

ウ 業務経験年数が5以上の者は、

仮合格か合格になる。

エ 経理の得点が60未満の者は、

不合格になる。

業務経験年数 ≥ 5	Y	Y	Y	N
3科目合計得点 ≥ 260	Y	Y	N	—
英語得点 ≥ 90	Y	N	—	—
合格	X	—	—	—
仮合格	—	X	—	—
不合格	—	—	X	X

解答解説

決定表に関する問題である。

アの英語の得点が90以上の場合は合格である。仮合格は誤りである。

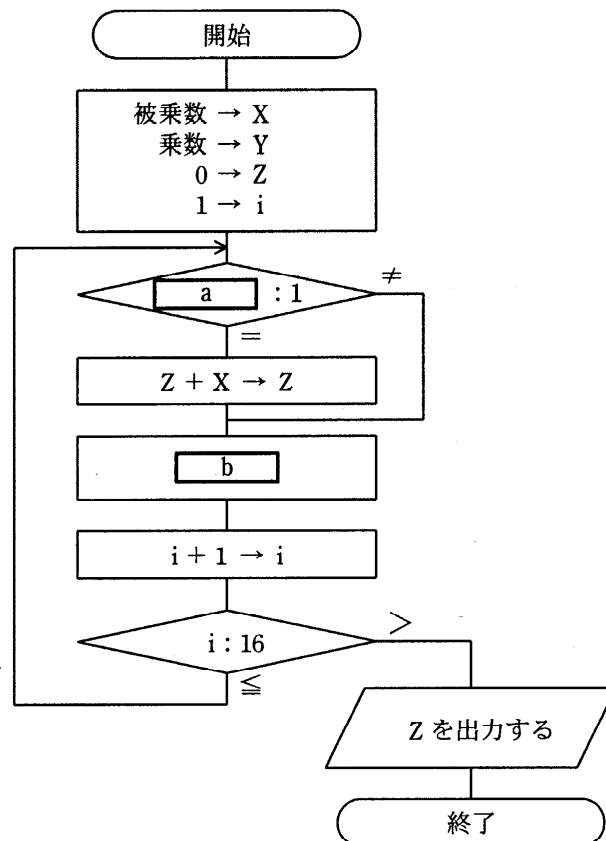
イの英語の得点が90未満の場合は仮合格になる。不合格は誤りである。

ウの業務経験年数5年以上は合格、仮合格、不合格の場合がある。

エの経理の得点が60未満は不合格になる。3科目の合計点が260以上が合格であるから、経理以外の2科目が100点で、経理が60未満だと不合格になる。求める答えはエとなる。

例題演習

次の流れ図は、シフト演算と加算の繰返しによって2進数の乗算を行う手順を表したものである。この流れ図中の a, b の処理の組合せとして、正しいものはどれか。ここで、乗数と被乗数は符号なしの16ビットで表される。X, Y, Zは32ビットのレジスタであり、けた送りには論理シフトを用いる。



	a	b
ア	Yの最下位ビット	Xを1ビット左シフト, Yを1ビット右シフト
イ	Yの最下位ビット	Xを1ビット右シフト, Yを1ビット左シフト
ウ	Yの最上位ビット	Xを1ビット左シフト, Yを1ビット右シフト
エ	Yの最上位ビット	Xを1ビット右シフト, Yを1ビット左シフト

解答解説

2進数の乗算に関する流れ図の問題である。

2進数の乗算の手順

- ① 乗数Yの最下位ビットが1の時は、乗算結果ZにXを加算する。乗数Yの最下位ビットが0の時は、何もしない。
- ② 乗数Yを1ビット右にシフトし、被乗数Xを左に1ビットシフトする。
- ③ $i + 1 \rightarrow i$ を求める。
- ④ $i \leq 16$ ならば、①に戻り、 $i > 16$ になると、結果を出力して終了する

aはYの最下位ビット、bはXを1ビット左シフトして、Yを1ビット右シフトする。求める答えはアとなる。

例題演習

整数Aを整数Bで割って余りを得るための関数 $\text{mod}(A, B)$ が次のように定義されているとき、関数呼出によって得られる値として正しいものはどれか。

[定義]

$\text{mod}(A, B)$ は、除数Bと同じ符号をとり、その絶対値はBの絶対値より小さい、適切な整数Nを選ぶことによって、

$$A = B \times N + \text{mod}(A, B)$$

を満足する。

ア $\text{mod}(11, 5) = 2$

イ $\text{mod}(11, -5) = -1$

ウ $\text{mod}(12, -5) = -3$

エ $\text{mod}(-12, 5) = 2$

解答解説

余りを求める関数 $\text{mod}(A, B)$ に関する問題である。

次の点に注意して処理を検討する必要がある。

- ① 余りは除数Bと同じ符号である。
- ② 余りの絶対値は除数Bの絶対値より小さい。
- ③ A、B、余りの間には次の関係が成立する。

$$A = B \times N + \text{mod}(A, B) \quad \text{但し、} N \text{は整数である}$$

この条件を利用して、解答群のア～エについて検討する。

アは $11 \div 5$ の余りを求める問題で、 $\text{mod}(11, 5) = 1$ であるから正しくない。

イは $11 \div (-5)$ の余りを求める問題で、①、②の条件から商は3、余りは-4となる。正しくない。

ウは $12 \div (-5)$ の余りで、商は3、余りは-3となる。正しい。求める答えはウとなる。

エは $(-12) \div 5$ の余りを求める問題である。①、②の条件から商は3で、余りは3となる。

$A = 5 \times (-3) + 3 = -12$ 。正しくない。

例題演習

pを2以上の整数とする。任意の整数nに対して、

$$n = k p + m \quad (0 \leq m < p)$$

を満たす整数kとmが一意に存在する。このmをnのpによる剰余といい、 $n \bmod p$ で表す。

$(-10000) \bmod 32768$ に等しくなるものはどれか。

ア $-(10000 \bmod 32768)$

イ $(-22768) \bmod 32768$

ウ $10000 \bmod 32768$

エ $22768 \bmod 32768$

解答解説

剰余に関する問題である。

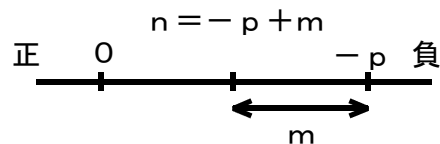
$n = k p + m$ で、 $0 \leq m < p$ であるから、 n が負の場合、 k が負で、 m の値が正になる答えとなる。

n の絶対値が p よりも小さい場合は次のようになる。

$n < 0$ の場合、 $k = -1$ であり、 $n = -p + m$ となり、 $m = n + p$

$n > 0$ の場合、 $k = 0$ であり、 $m = n$

$$(-10000) \bmod 32768 = -10000 + 32768 = 22768$$



ア～エについて、余りを求めると

アの場合、 -10000

イの場合、 $-22768 + 32768 = 10000$

ウの場合、 10000

エの場合、 22768 となる。求める答えはエとなる。

例題演習

アルファベット3文字で構成されるキーがある。次の式によってハッシュ値 h を決めるとき、キー“SEP”と衝突するのはどれか。ここで、 $a \bmod b$ は、 a を b で割った余りを表す。

$$h = (\text{キーの各アルファベットの順位の総和}) \bmod 27$$

アルファベット	順位	アルファベット	順位	アルファベット	順位
A	1	J	10	S	19
B	2	K	11	T	20
C	3	L	12	U	21
D	4	M	13	V	22
E	5	N	14	W	23
F	6	O	15	X	24
G	7	P	16	Y	25
H	8	Q	17	Z	26
I	9	R	18		

ア APR

イ FEB

ウ JAN

エ NOV

解答解説

ハッシュ法の衝突に関する問題である。

$$SEP = 19 + 5 + 16 = 40$$

$$40 \bmod 27 = 13$$

ア～エについてハッシュ値を求める。

$$\text{アのAPR} = 1 + 16 + 18 = 35$$

$$35 \bmod 27 = 8$$

$$\text{イのFEB} = 6 + 5 + 2 = 13$$

$$13 \bmod 27 = 13$$

$$\text{ウのJAN} = 10 + 1 + 14 = 25$$

$$25 \bmod 27 = 25$$

$$\text{エのNOV} = 14 + 15 + 22 = 51$$

$$51 \bmod 27 = 24$$

SEPと衝突するのはFEBであり、求める答えはイとなる。

例題演習

与えられた正の整数 $x_0, x_1 (x_0 > x_1)$ の最大公約数を、次の手順で求める。 $x_0 = 175$, $x_1 = 77$ の場合、手順(2)は何回実行するか。ここで、“ $A \rightarrow B$ ” は、 A を B に代入することを表す。

〔手順〕

- (1) $2 \rightarrow i$
- (2) x_{i-2} を x_{i-1} で割った剰余 $\rightarrow x_i$
- (3) $x_i = 0$ ならば x_{i-1} を最大公約数として終了する。
- (4) $i + 1 \rightarrow i$ として(2)に戻る。

ア 3

イ 4

ウ 6

エ 7

解答解説

最大公約数を求めるアルゴリズムの問題である。

$x_0 = 175$ 、 $x_1 = 77$ であるから、 $x_2 = 21$ となる。

2 回目の計算は、 $x_0 = 77$ 、 $x_1 = 21$ であるから、 $x_2 = 14$ となる。

3 回目の計算は、 $x_0 = 21$ 、 $x_1 = 14$ であるから、 $x_2 = 7$ となる。

4 回目の計算は、 $x_0 = 14$ 、 $x_1 = 7$ であるから、 $x_2 = 0$ となる。

答えは 4 となり、求める答えはイとなる。

① 再帰アルゴリズム(リカーシブ)

② リカーシブとは

プログラムの中から自分自身を呼び出すことを再帰呼出という。自分自身を定義するのに自分自身よりも1次低い集合を用いる。その部分集合はより低次の部分定義を用いて定義することを繰り返して表現する。再帰呼出には再帰からの脱出口がなければならない。脱出口がない場合、再帰呼出は永遠に続くことになる。再帰呼出を行うときは、呼んだ自分自身から制御が戻ってきたときに、前の状態で引き続き実行できるように、実行アドレスや引数、使用していた変数などの情報を保管しておく。そのための保管領域として、後入れ先出しの特徴を持つスタックを使用する。

③ 再帰呼び出しの具体例

フィボナッチの数列

1 1 2 3 5 8 13 21 34 55 89

フィボナッチの数列の定義

$$F_n = F_{n-1} + F_{n-2} \quad n \geq 3$$

$$F_1 = F_2 = 1 \quad n = 1 \text{ または } 2$$

フィボナッチの数列は再帰呼び出しを使用して処理ができる。定義した関数 F_n をスタックに格納し、 F_n の中で使用している F_{n-1} を次に格納し、順次1次低い定義関数を格納していくと、最後に既知の値 $F_2 = 1$ 、 $F_1 = 1$ にたどり着き、再帰呼び出しから脱出して、格納した順序とは逆に各関数の値を求めることができる。

スタックを利用した解答手順

㊦ 次の各式を逐次スタックにPUSHする。

$$F_n = F_{n-1} + F_{n-2}$$

$$F_{n-1} = F_{n-2} + F_{n-3}$$

⋮

$$F_4 = F_3 + F_2$$

$$F_3 = F_2 + F_1$$

㊧ $F_2 = F_1 = 1$ であるから、 $F_2 + F_1 = 2$ となり、再帰呼出しから脱出する。

㊨ スタックの各式をPOPして値を求める。

$$F_3 = F_2 + F_1 = 2$$

$$F_4 = F_3 + F_2 = 3$$

⋮

$$F_{n-1} = F_{n-2} + F_{n-3}$$

$$F_n = F_{n-1} + F_{n-2} \text{ を利用して } F_n \text{ を求めることができる。}$$

② バッカス記法

① バッカス記法とは

バッカス記法は言語を定義するための言語である。文章をどのようにして構成するかという構文規則を示している。

バッカス記法の「::=」や「|」を超記号と呼ぶ。「<」と「>」で囲まれたものを構文要素と呼び、これを構文則という。

② 構文則によって定義される文章

<文>から始まる。構文要素をその右辺のものと置き換えて、構文要素を一つも含まなくなったものである。

③ 表記法

⑦ <数字>::=0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

⑧ <英字>::=A | B | C | D | E | F | G | H | I | J | K | L | M | N |
O | P | Q | R | S | T | U | V | W | X | Y | Z

⑨ <名前>::=<英字> | <名前><英字> | <名前><数字>

④ 表記法の内容

⑦は「<数字>は0～9のどれかである」

⑧は「<英字>はA～Zのどれかである」。「|」は「または」という意味である。

⑨の最初の<名前>::=<英字>は「英字は名前である」と読む。

Aという英字は名前として使うことができる。

| <名前><英字>は「または、名前の次に英字をつけたものも名前である」と読む。

Aは名前であるから、その次に英字AをつけたものAAも名前である。AAは名前であるから、その次に英字YをつけたものAAYも名前である。

| <名前><数字>は「または、名前の次に数字をつけたものも名前である」と読む。Bは名前であるから、その次に数字をつけたB3は名前である。B3は名前であるから、その次に数字9をつけたものB39も名前である。

名前は英字一つまたは英字で始まりその後に英字又は数字をいくつかつけたものである。

⑤ バッカス記法の具体例

日本語の文章は主部と述部からなり、主部は名詞と助詞からなり、述部は動詞からなるとする。名詞としては、私、君、犬、助詞としては、は、が、も、動詞としては、遊ぶ、泳ぐ、走

る、があるものとし、これをバックス記号で表すと次のようになる。

<文> ::= <主部> <述部>
<主部> ::= <名詞> <助詞>
<述部> ::= <動詞>
<名詞> ::= 私 | 君 | 犬
<助詞> ::= は | が | も
<動詞> ::= 遊ぶ | 泳ぐ | 走る

「私は遊ぶ」という文章

<文> ::= <主部> <述部> | <名詞> <助詞> <述部> |
 <名詞> <助詞> <動詞> | 私 <助詞> <動詞> | 私は <動詞> | 私は遊ぶ
「私」、「は」、「遊ぶ」を終端記号、構文要素を非終端記号という。

③ 正規表現・分割統治法

④ 正規表現とは

正規表現は、字句をパターンの集合で表す式で定義する表現方法である。文字列に対し、メタキャラクタといわれる特殊な記号を使用し、組み合わせて任意の文字列を表現する。

⑤ 正規表現の代表的なメタキャラクタ

主な正規表現の具体例を表に示す。

メタキャラクタ	意味	例
^	行の先頭	^ A 行頭がAの文字で始まる
\$	行の末尾	Z \$ 行末がZの文字で終わる
.	任意の1文字	M. . N Mで始まりNで終わる4文字
?	任意の1文字	M??N Mで始まりNで終わる4文字
[]	[]内に含まれる任意の1文字	[A-Z] A~Zの範囲の文字
*	直前文字の0回以上の繰返し	A B X * Xを0回以上繰返す
+	直前文字の1回以上の繰返し	A B X + Xを1回以上繰返す

⑥ 分割統治の考え方

分割統治法は、解くべき問題を小規模な問題に分割し、各部分問題の解を結合することによって、全体の解を求めようとする解決法である。分割というのは、対象とする変数の集合をいくつかに分けたり、定義領域を分けることであり、多くの場合、再帰的に反復して利用する。

④ 分割統治法の応用

クイックソートやマージソートは分割統治の考え方を利用したアルゴリズムであり、システム分析やプログラム設計にもこの考え方が用いられている。

モジュール分割は大きなプログラムを主たる部分とデータ入出力やデータチェック部分に分割し、各モジュールを個別に作成し、テストし、最後に各モジュールを結合して完全なプログラムを作成する。この考え方は分割統治法の応用である。

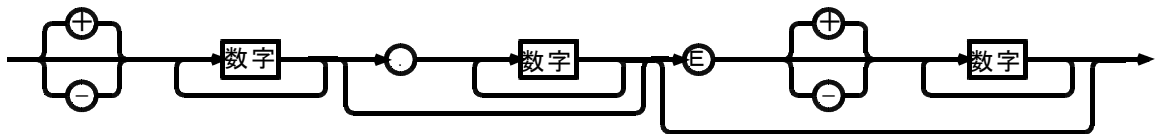
例題演習

第1図のような記述法による構文図を考える。－1、21.5、+5.23などの表現は、この構文図の規定に合致する。この記述法に従うとき、第2図の規定に合致する数値表現はどれか。

第1図



第2図



ア －52.3E05

イ .58E2

ウ +.90E11

エ 3.45E

解答解説

構文図に関する問題である。

第1図を利用して基本的な考え方を理解し、その考え方を利用して、第2図の構文図の内容を読み取る問題である。

アは構文図の規定に合致している。求める答えはアとなる。

イは小数点の前に数字がない。

ウは+と小数点の間に数字がない。

エはEの後に数字がない。

例題演習

次のBNFで定義されるビット列Sであるものはどれか。

$\langle S \rangle ::= 01 \mid 0 \langle S \rangle 1$

ア 000111

イ 010010

ウ 010101

エ 011111

解答解説

B N Fに関する問題である。

0 1、または $0 < S > 1$ であるから、0と1の間に0 1が挟まるビットパターンになる。

アは0 1の間に0 1が挟まるビットパターンである。求める答えはアとなる。

イは0と0の間に0 1が挟まるパターンになっており正しくない。

ウは0と1の間に1 0が挟まるパターンであり正しくない。

エは0と1の間に1 1が挟まるパターンであり正しくない。

例題演習

数値に関する構文が次のとおり定義されているとき、＜数値＞として扱われるものはどれか。

＜数値＞ : : = <数字列> | <数字列> E <数字列> | <数字列> E <符号> <数字列>

＜数字列> : : = <数字> | <数字列> <数字>

＜数字> : : = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

＜符号> : : = + | -

ア -12

イ 12E-10

ウ +12E-10

エ +12E10

解答解説

構文に関する問題である。

ア、ウ、エは先頭に符号があるため定義された構文ではない。

イは＜数字列> E <符号> <数字列>の構文を満たすため答えとなる。求める答えはイとなる。

例題演習

次の表は、入力文字列を検査するための状態遷移表である。この検査では、初期状態を a として、文字列の入力中に状態が e になれば不合格とする。解答群で示される文字列のうち、この検査で不合格となるものはどれか。ここで、解答群中の△は空白を示す。

		入力文字				
		空白	数字	符号	小数点	その他
現状 の 状態	a	a	b	c	d	e
	b	a	b	e	d	e
	c	e	b	e	d	e
	d	a	e	e	e	e

ア +0010

イ -1

ウ 12.2

エ 9.△

解答解説

状態遷移表を利用した文字列検査に関する問題である。

ア～エについて、状態遷移表を使用して遷移を実行すると次のようになる。

アの場合、 $a \rightarrow c \rightarrow b \rightarrow b \rightarrow b \rightarrow b$

イの場合、 $a \rightarrow c \rightarrow b$

ウの場合、 $a \rightarrow b \rightarrow b \rightarrow d \rightarrow e$ 不合格となり、求める答えはウとなる。

エの場合、 $a \rightarrow b \rightarrow d \rightarrow a$

例題演習

次の規則から生成することができる式はどれか。

〔規則〕

$\langle \text{式} \rangle ::= \langle \text{変数} \rangle \mid (\langle \text{式} \rangle + \langle \text{式} \rangle) \mid \langle \text{式} \rangle * \langle \text{式} \rangle$

$\langle \text{変数} \rangle ::= A \mid B \mid C \mid D$

ア $A + (B + C) * D$

イ $(A + B) + (C + D)$

ウ $(A + B) * (C + D)$

エ $(A * B) + (C * D)$

解答解説

正規表現に関する問題である。

次の規則から生成される。

① 式はA、B、C、Dの4文字のいずれかで構成される。

② ()の中の式は、式と式を+で結合したものである。

③ ()なしの場合、又は()の外は、式と式は*で結合される。

アの場合、()なしであるのに式が+で結合されている。

イの場合、()の外で式と式が+で結合されている。

ウの場合、生成規則①～③を満足している。求める答えはウとなる。

エの場合、()の中で、式と式が*で結合されている。

例題演習

自然数 n に対して、次のとおり再帰的に定義される関数 $f(n)$ を考える。 $f(5)$ の値はどれか。

$f(n) : \text{if } n \leq 1 \text{ then return } 1 \text{ else return } n + f(n - 1)$

ア 6

イ 9

ウ 15

エ 25

解答解説

再帰関数の値を求める問題である。

関数は $n \leq 1$ ならば1、そうでないならば $n + f(n - 1)$ であるから、次のようになる。

$$f(5) = 5 + f(4) = 5 + 4 + f(3) = 5 + 4 + 3 + f(2) = 5 + 4 + 3 + 2 + f(1) \\ = 5 + 4 + 3 + 2 + 1 = 15$$

答えは15で、求める答えはウとなる。

例題演習

正三角形の内部の点から、各辺に下ろした垂線の長さの和は一定である(図1参照)。三角グラフは、この性質を利用して、三つの辺に対応させた要素の割合を各辺への垂線の長さとして表したグラフである。図2の三角グラフは、3種類のソフトについて、A～Dの4人の使用率を図示したものである。正しい解釈はどれか。

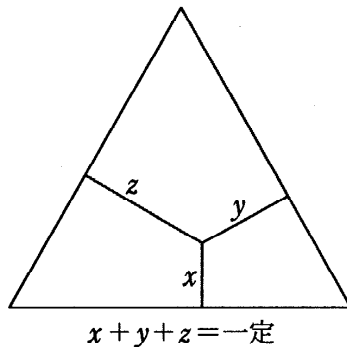


図1 正三角形の性質

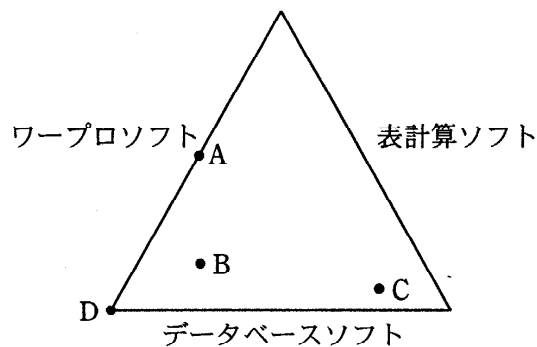


図2 三角グラフ

- ア Aさんは、ワープロソフトだけを使用している。
- イ Bさんは、ほかのソフトに比べて表計算ソフトの使用率が高い。
- ウ Cさんは、データベースソフト、表計算ソフト、ワープロソフトの順に使用率が高い。
- エ Dさんは、表計算ソフトを使用していない。

解答解説

三角グラフの使用法に関する問題である。

A、B、C、Dの4人のワープロソフト、表計算ソフト、データベースソフトの使用率が問題になっている。使用率はそれぞれの点から関係するソフトの辺に下した垂線の長さで決まる。各辺への垂線の長さの和は一定であるから、これを100として解釈すればよい。

アのAはワープロだけ使用しているは、表計算とデータベースを使用し、ワープロは使用していない。

イの表計算ソフトの使用率が高いは、点Bから表計算ソフトの辺への垂線が、他の辺への垂線の長さよりも長いから正しい。求める答えはイとなる。

ウのCの使用率の高い順は、ワープロソフト、表計算ソフト、データベースソフトの順となる。

エの表計算ソフトだけを使用している。