

文字列の照合

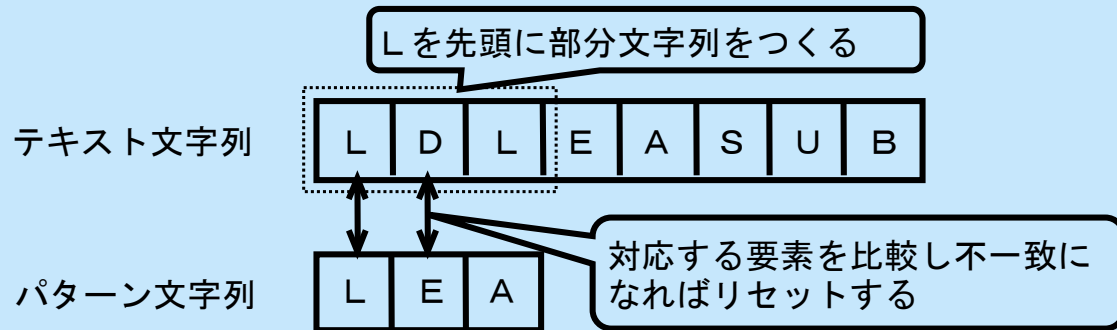
力任せの探索法とは

① 力任せの探索

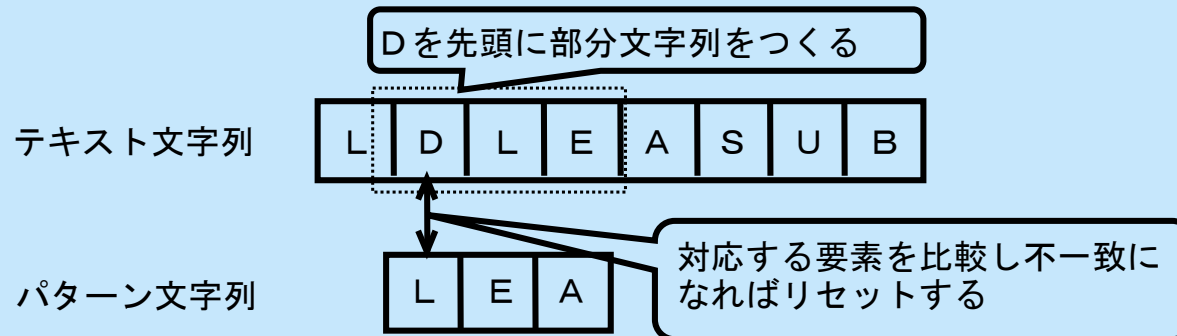
- ① テキスト文字列の中に
- ② パターン文字列と同じ長さの部分文字列をつくり、
- ③ パターン文字列の対応する文字と順次比較し、
- ④ 全文字が一致すれば探索成功である。
- ⑤ 比較途中で不一致になると、
配列の次の要素に移動して、
- ⑥ 新しい部分文字列を作りパターン文字列と比較する。

- ② 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。

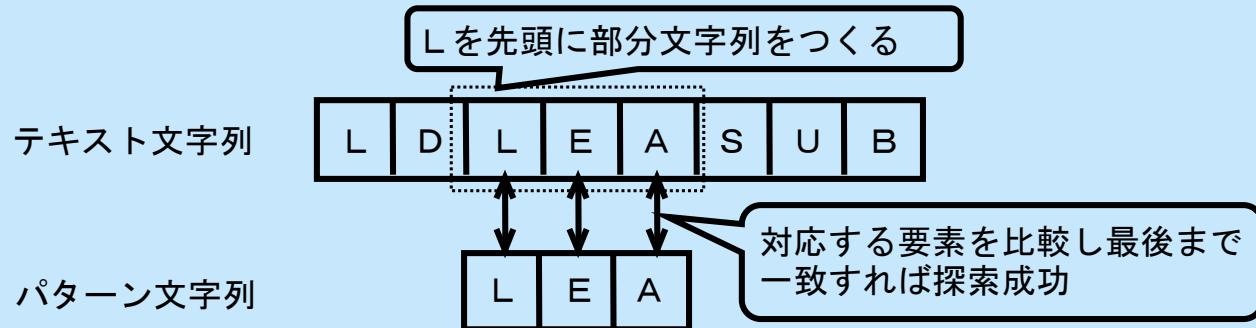
最初の比較



2回目の比較



3回目の比較(探索成功)



力任せの探索法の手順

① ポインタの初期化

- ① 一つのポインタ i をテキスト文字列(長さ n)上に、
- ② 一つのポインタ j をパターン文字列上に置く。
- ③ i および j を1で初期化する。

② テキスト文字列上に部分文字列を作成

- ① テキスト上にポインタ i を先頭に
- ② 長さ m ($m < n$)の部分文字列を作成する。
- ③ 部分文字列の要素の添字を $i + j - 1$ で表す。

③ 添字jを操作して文字列を比較する。

- ① テキスト上の部分文字列と
パターン文字列の対応する要素を比較する。
- ② 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う。
- ③ 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。
- ④ 対応する要素の内容が
比較途中で不一致になると④に移る。

- ④ 比較途中で不一致になるか
または照合が得られた場合、
- ① パターンの添字 j をリセットし、
 - ② テキスト文字列上のポインタ i を
インクリメントする。
 - ③ ②に戻る。
- ⑤ 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。

力任せの探索法の比較回数

- ① テキスト長 N 、パターン長 M とすると、
最悪の比較回数は、次の式で求まる。

$$\text{最悪の比較回数} = (N - M + 1) \times M$$

- ② テキスト長 N の値が、パターン長 M の値に比べて、
大きい場合は NM となる。

ボイヤ・ムーアの文字列探索法

① ボイヤ・ムーア

- ① 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
- ② かなり速い文字列探索法を開発することができる。

- ② 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。

- ③ 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。

ボイヤ・ムーアの比較回数

- ① 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。
- ② 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。
- ③ 平均的な比較回数は N/M となる。

- ④ 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。
- ⑤ 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。
- ⑥ このアルゴリズムは最悪の場合は線形となる。

移動距離の計算要領

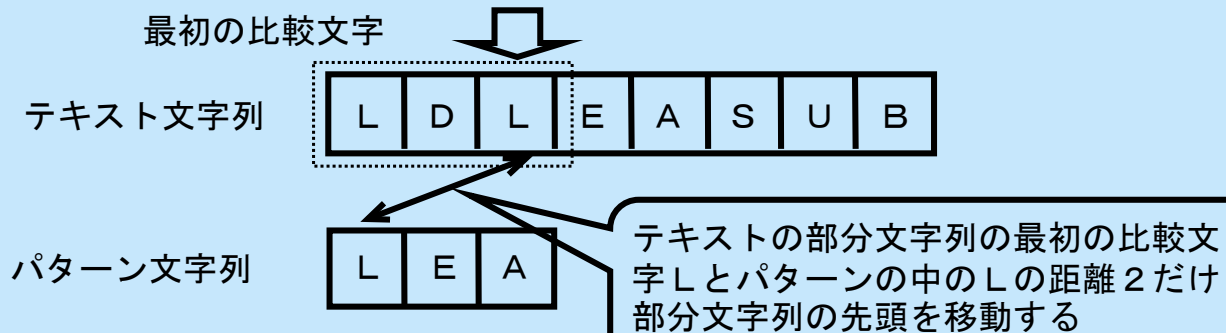
- ① 部分文字列の後方から
テキストとパターンを比較する場合、
 - ① テキストの最初の比較文字が
パターンの比較の最後にある場合、
 - ② パターンの中に存在しない場合、
 - ③ パターン長 m だけ、
部分文字列の先頭位置を移動させる。

- ② テキストの最初の比較文字が
パターンの中にある場合、
 - ① 二つの文字の移動距離を求め、
 - ② 移動距離分だけ
部分文字列の先頭位置を移動する。

- ③ テキストの最初の比較文字が
パターンの中に複数個ある場合、
 - ① 最後尾の文字までの移動距離を使用する。

移動距離計算例1

① 移動距離計算例1の図

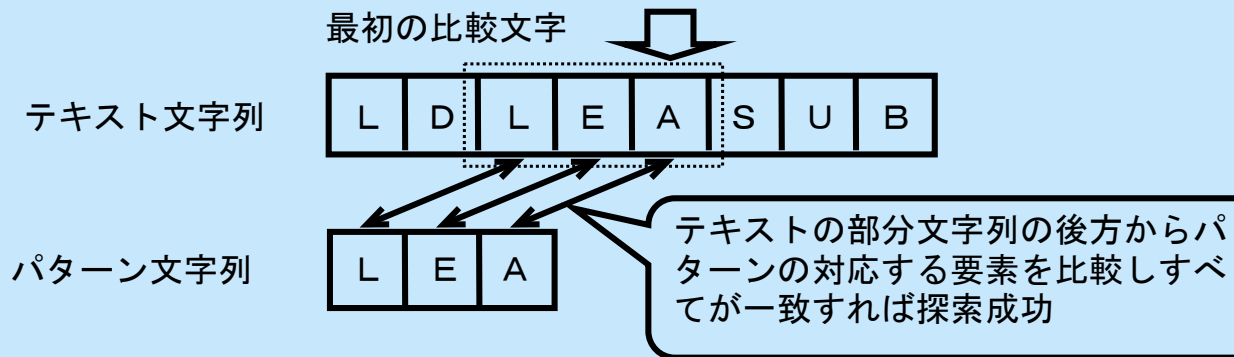


② 移動距離計算例1の説明

- ① テキストの部分文字列LDLと
パターン文字列LEAを後方から比較する。
- ② 最初の比較文字Lは、
パターンの先頭のLと一致する。
- ③ 計算要領の②に相当する。
- ④ パターン文字列の移動距離は2となる。
- ⑤ 次の部分文字列はLEAとなる。一致する。

移動距離計算例2

① 移動距離計算例2の図



② 移動距離計算例2の説明

- ① 次のテキストの部分文字列はEASである。
- ② テキストの最初の比較文字はSとなる。
- ③ Sはパターンの中に存在しないため、
- ④ 部分文字列の先頭位置は3要素分移動し、
- ⑤ テキスト文字列の範囲外となる。
- ⑥ この場合の移動距離は3となる。

文字列探索の要領

- ① 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。
- ② 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。
- ③ 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。

- ④ $T(i) \neq P(j)$ となると、
照合を中断して、照合ループから脱出する。
- ⑤ 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。
- ⑥ 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。
- ⑦ 次の照合の開始点を $i + d(T(i))$ を求める。

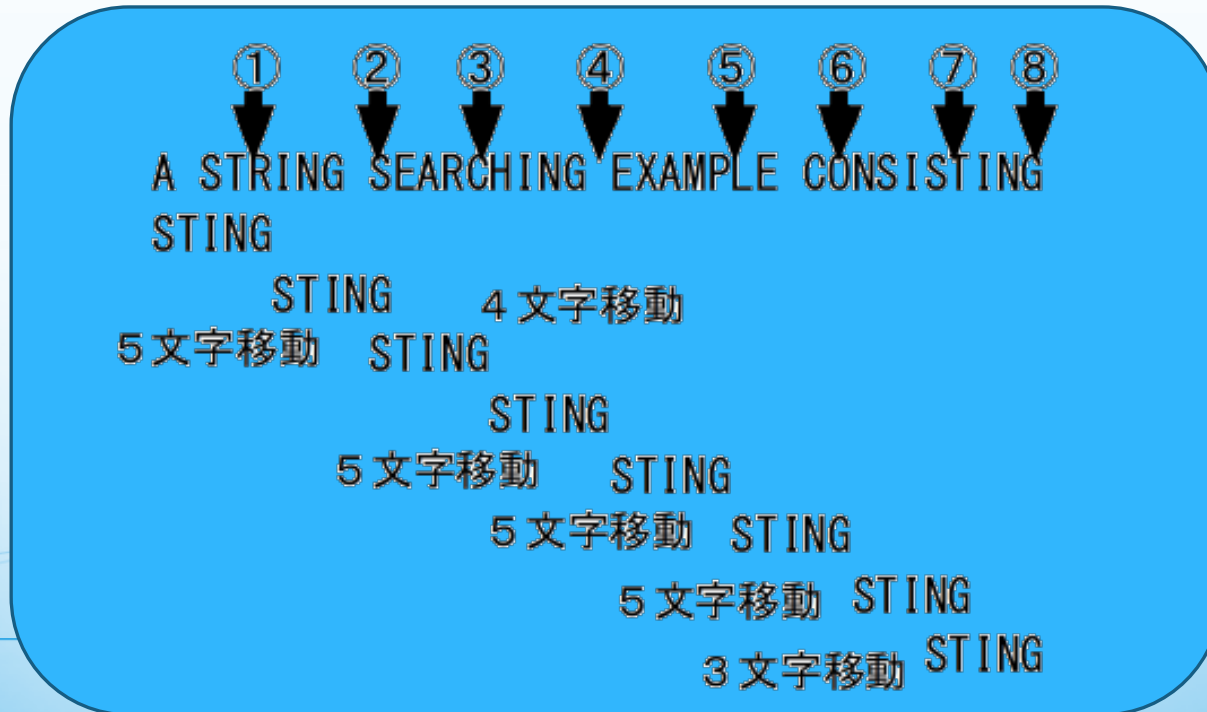
⑧ 照合を繰り返し、 $j \leq 0$ となると、

- ① 照合ループから脱出し、 $1 \rightarrow \text{found}$ として、
- ② 更に $i+1 \rightarrow i$ を求めて、
- ③ パターン文字列と一致する
テキスト上の位置を決め、
- ④ 処理を終了する

⑨ ⑦で新しく求めた開始点を起点として、
③～⑧の操作を繰り返す。

⑩ ③～⑨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。

パターン長5(STING)の場合の具体例



パターンSTRINGの操作要領

- ① 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。



テキスト
パターン

A STRING SEARCHING EXAMPLE CONSISTING
STRING

- ② 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、⑪の操作に移る。



テキスト A STRING SEARCHING EXAMPLE CONSISTING
パターン STING



テキスト A STRING SEARCHING EXAMPLE CONSISTING
パターン STING

- ③ 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。
- ④ 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。



テキスト
パターン

A STRING SEARCHING EXAMPLE CONSISTING
STRING

- ⑤ 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。



テキスト
パターン

A STRING SEARCHING EXAMPLE CONSISTING
STRING

4文字右に移動する前

- ⑥ 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。



テキスト
パターン

A STRING SEARCHING EXAMPLE CONSISTING
STING

4文字右に移動させた結果

⑦ 次に、テキスト上文字Cと
パターン文字列を比較する。

- ① 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。
- ② そこで、パターン文字列を右に5文字分移動させる。
- ③ 以下、同じことを繰り返す。

A STRING SEARCHING EXAMPLE CONSISTING
↓
STRING

⑧ 同様の移動を3回行くと、
CONSISTINGのTに到着する。

A STRING SEARCHING EXAMPLE CONSISTING
STING


A STRING SEARCHING EXAMPLE CONSISTING
STING

A STRING SEARCHING EXAMPLE CONSISTING
STING

⑨ このテキスト上の文字Tが

- ① 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、
- ② 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。

A STRING SEARCHING EXAMPLE CONSISTING
STRING



- ⑩ テキスト上の文字Gが
パターンの右端の文字Gと一致する。

- ⑪ 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、
 - ① それぞれの文字列の比較対象文字を
 - ② 一つ前の文字に移動し、
 - ③ 二つの文字の比較を繰り返す。

- ⑫ テキストの文字列とパターンの文字列が
完全に一致すると、探索成功になる。

⑬ 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、

- ① 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、②の処理を行う。
パターン上になければ、③の処理を行う。
- ② 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。
- ③ 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。

特殊なパターン文字列の照合1

テキスト文字列 AABAAABAAAAB

パターン文字列 AAAA

テキスト文字列がAABAAABAAAAB、パターン文字列が4文字AAAAでできている場合の処理手順について考える。

- ① テキスト文字列の
最初の4文字とパターン文字列を比較する。

テキスト文字列 AABAAABAAAAB
パターン文字列 AAAA

- ② テキスト文字列の3文字目で不一致になり、
この間の比較は無意味である。

- ③ パターン文字列の先頭を
テキストの文字列の4番目まで移動できる。

テキスト文字列 AABAAABAAAAB
パターン文字列 AAAA

- ④ テキスト文字列の
7番目で不一致になり、
それまでの比較は無意味となる。

テキスト文字列 AABAAABAAAAB
パターン文字列 AAAA

- ⑤ パターン文字列が同じ文字列の場合、
途中で不一致になると、
テキスト文字列の不一致になった
次の文字までパターン文字列の先頭を
移動させることができる。

特殊なパターン文字列の照合2

テキスト文字列 AABAAABAAAAB

パターン文字列 AABB

パターン文字列がAABBの場合について考える。

- ① パターン文字列の移動は次のようになる。

テキスト文字列 AABAAABAABB

パターン文字列 AABB

AABB

AABB

AABB

- ② 不一致になったパターンの文字が、
パターン文字列の先頭文字と異なる場合、
テキスト文字列の不一致になった位置まで、
パターン文字列の先頭を移動させることができる。