

ヒーブホ

# ヒープとは

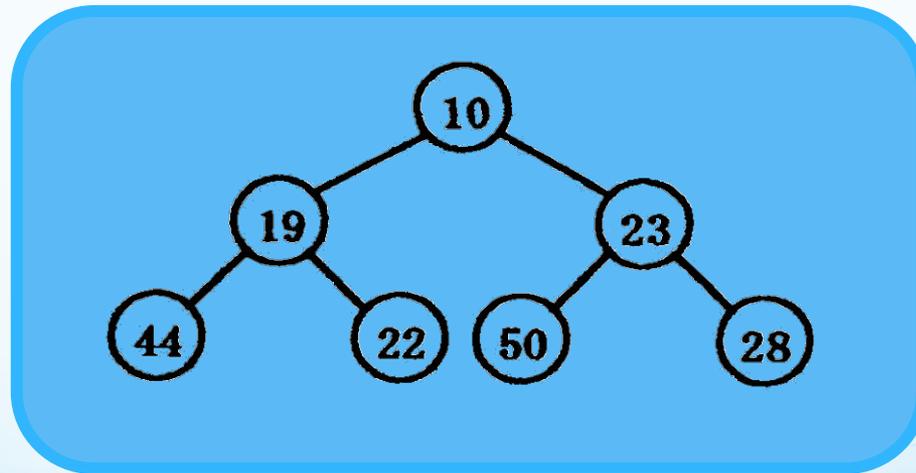
- ① 完全二分木の各節に一連番号を付け、その番号をもとに、完全二分木のデータ構造を配列で表現したものである。
- ② 二分木に番号をつけ、どの親子の値をとってみても次の関係になっている配列である。  
「親の値 < 子の値」または「親の値 > 子の値」

# 一連番号を付け方

- ① 根には番号1をつける。
- ② 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。
- ③ 親の節の値と子の節の値を次のいずれかにする。
  - ① 親の節の値は子の節の値よりも小さい
  - ② 親の節の値は子の節の値よりも大きい

- ④ この番号づけによって、  
子の節番号、親の節番号を  
容易に求めることができる。
- ⑤ 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。
- ⑥ 逆にすれば最大になる。
- ⑦ ヒープ木の節に、  
一連番号付けの規則に従って番号をつけ、  
この節番号を配列の要素番号とする要素に  
格納して配列を作ることができる。

# ヒープ木と配列



要素番号	1	2	3	4	5	6	7
節の値	10	19	23	44	22	50	28

# ヒープの探索

- ① ヒープの探索は線形探索と同じである。
- ② 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する。
- ③ 各要素の値と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。

# 親の値 $\leq$ 子の値の場合の挿入手順

- ① 挿入する要素を最後部へ無条件に追加する。
- ② 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。
- ③ 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。

- ④ この比較・交換を根に向かって行い、  
次の関係になるまで続ける。

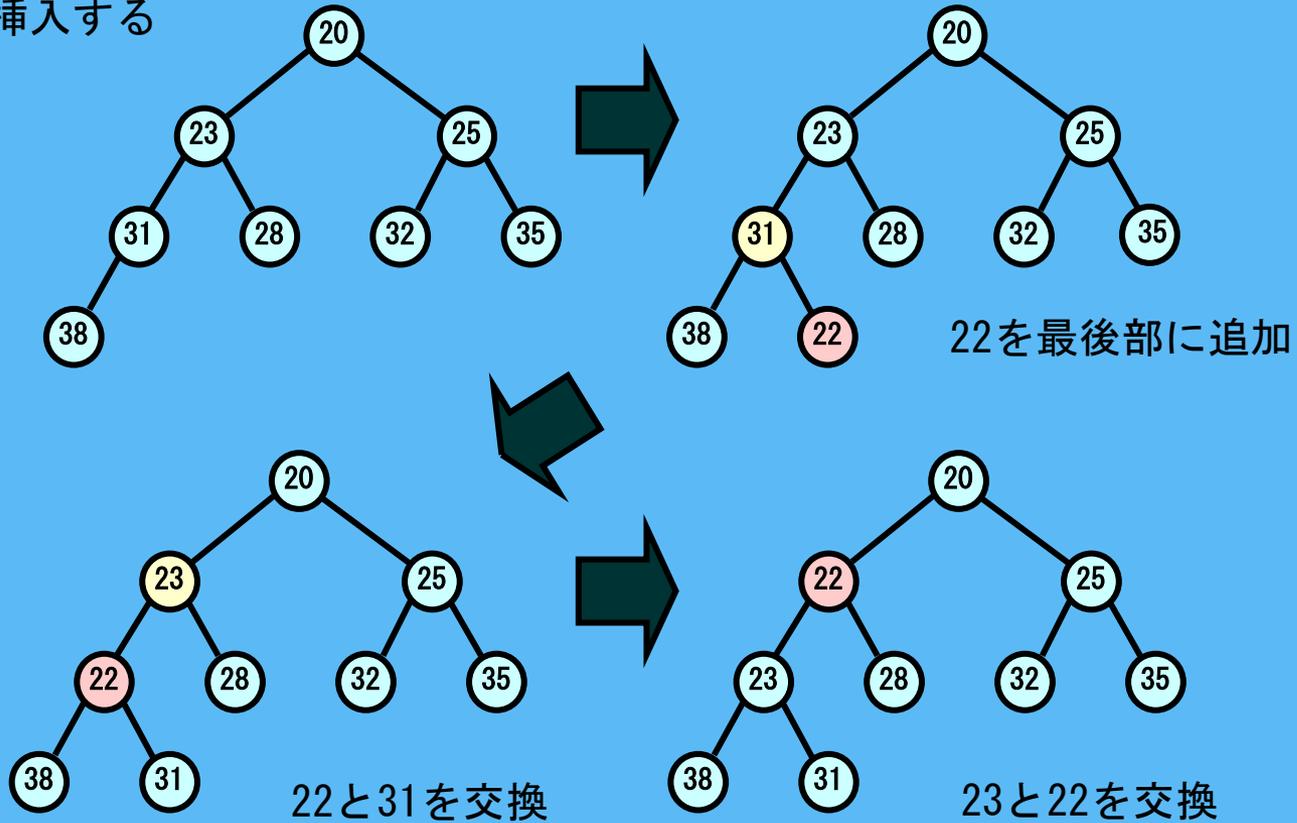
「親の値  $\leq$  子の値」

- ⑤ ヒープ木に値22のデータを挿入する具体例を  
次のページに示す。

挿入の手順は矢印の方向に進める。

# ヒープ挿入の具体例

22を挿入する



# ヒープの削除とは

- ① ヒープの削除は、  
次のいずれかの処理を行うことである。
  - ① 最小の値の要素を取り出す
  - ② 最大の値の要素を取り出す
- ② 最小の値の要素または最大の値の要素は、  
ヒープの性質上、根にある。
- ③ ヒープの削除は根の削除に相当する。

# 親の値 $\leq$ 子の値の場合の削除手順

- ① 根の部分の値を削除する。
- ② 根の部分にヒープの最後の要素を移動する。
- ③ 根から順次、次の条件が保たれるように要素の値を比較・交換する。

親の値  $\leq$  子の値

- ④ ヒープ木のすべての親子関係が  
次の条件になると、新しいヒープ木となる。

親の値  $\leq$  子の値

- ⑤ ヒープ木から値20のデータを削除する具体例を  
次のページに示す。

削除の手順は矢印の方向に進められる。

# ヒープの削除の具体例

20を削除する

