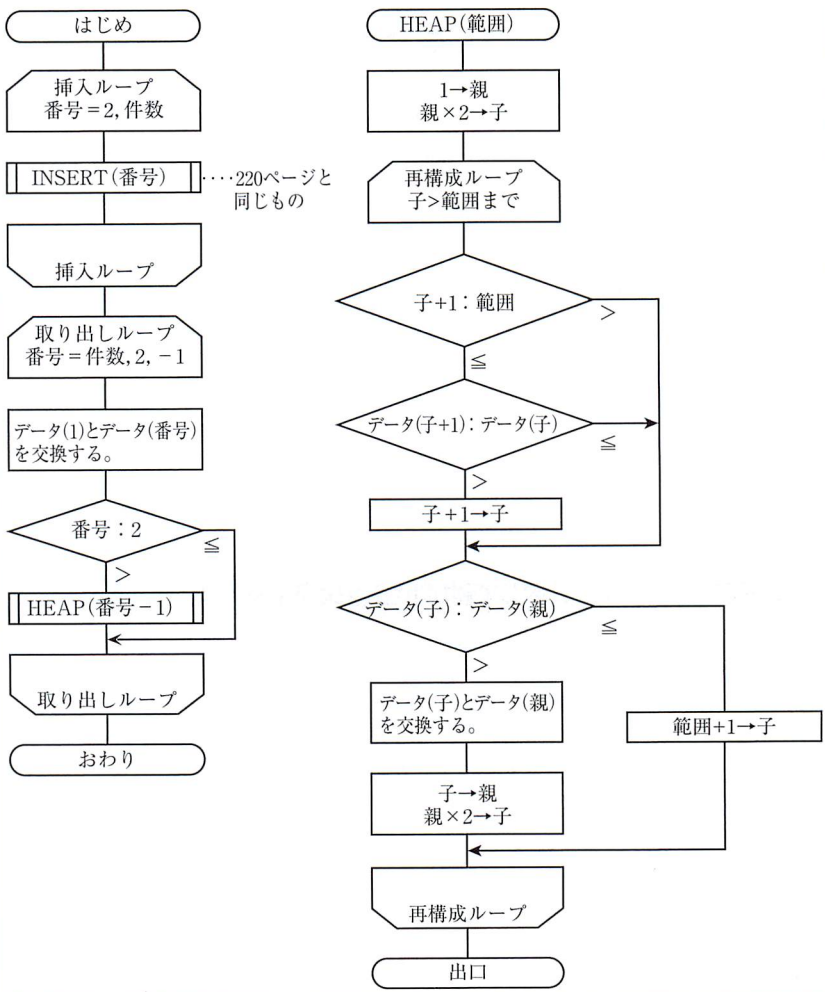
23 ヒープソート

1 アルゴリズムの概要

- ヒープの根から最大値を取出し，配列の末尾に置くことを，ヒープを再構成しながらくり返し，データを整列する。

2 流れ図



3 整列の様子

件数＝8　　データ

20	50	10	30	70	40	60	80

①挿入ループ

- 222ページで示したとおり，ヒープを作る。
- 挿入ループ終了後の配列の内容は，次のようになる。

80	70	60	50	30	10	40	20

②取出しループのヒープの再構成

- 根（番号＝1）のデータと末尾のデータを交換する。つまり，末尾に，最大値が設定される。
- 親＞子となるようにヒープを再構成する。

範囲	親	子	交換	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	
7	1	2	○	70	20	60	50	30	10	40	80	図①
	2	4	○	70	50	60	20	30	10	40	80	
6	1	3	○	60	50	40	20	30	10	70	80	図②
	3	6	×	60	50	40	20	30	10	70	80	
5	1	2	○	50	10	40	20	30	60	70	80	図③
	2	5	○	50	30	40	20	10	60	70	80	
4	1	3	○	40	30	10	20	50	60	70	80	図④
3	1	2	○	30	20	10	40	50	60	70	80	図⑤
2	1	2	○	20	10	30	40	50	60	70	80	図⑥

③根を取出して整列終了

- データが2個になったら，根（番号＝1）と末尾（番号＝2）のデータを交換するだけで整列が終わる。

（注）図は次のページに示す。

10	20	30	40	50	60	70	80

1 最大値が容易に選択できるヒープソート

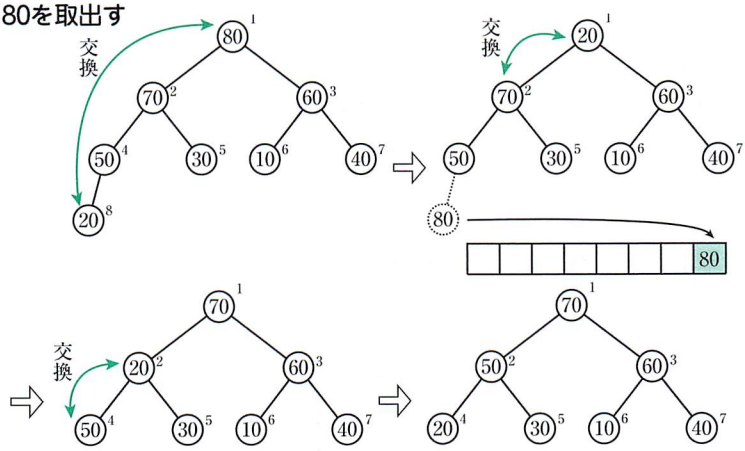
最大値を選び出して，配列の末尾から順に置いていくことで整列するのが基本選択法でした。**ヒープソート**は，ヒープの根に設定されている最大値を取出すことで，最大値を求める時間を選択法よりも短縮しています。実際には，根のデータと末尾のデータを交換することで，末尾に最大値を設定します。

その後，ヒープを再構成しなければなりません。しかし，ヒープは，データ数が2^nでも，たかだかnの深さしかなく，再構成も容易です。

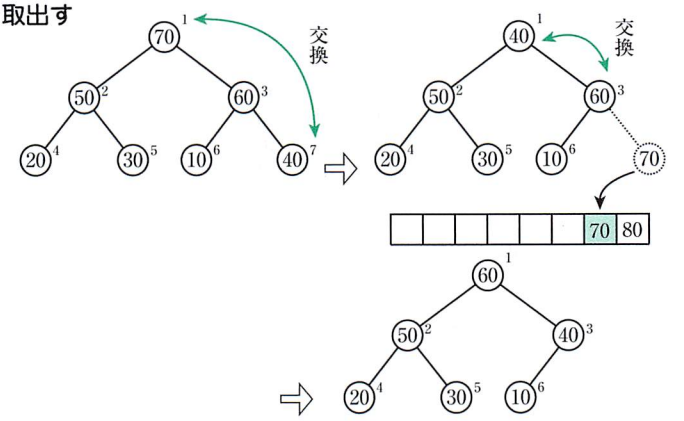
Point ヒープの根を取出すだけで，最大値を設定できます。

4 ヒープ再構成の様子

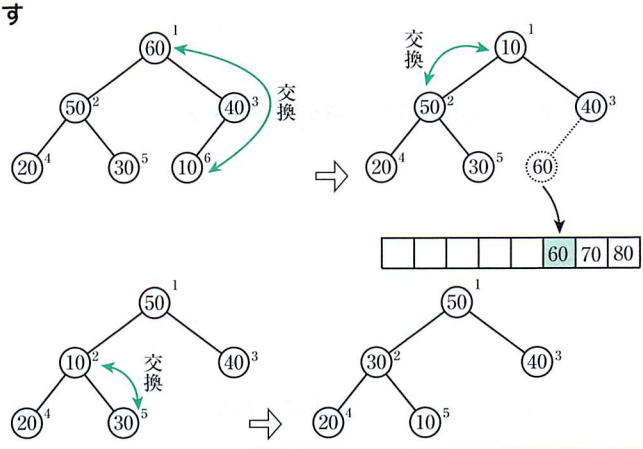
図①　80を取出す



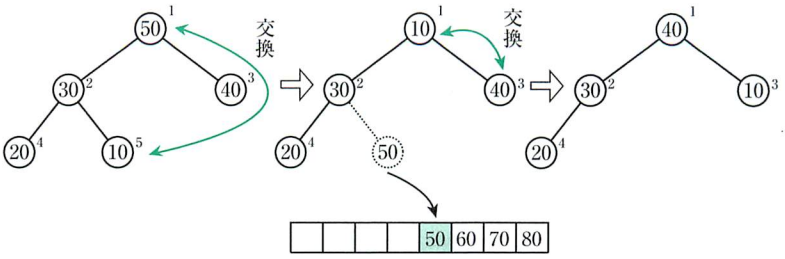
図②　70を取出す



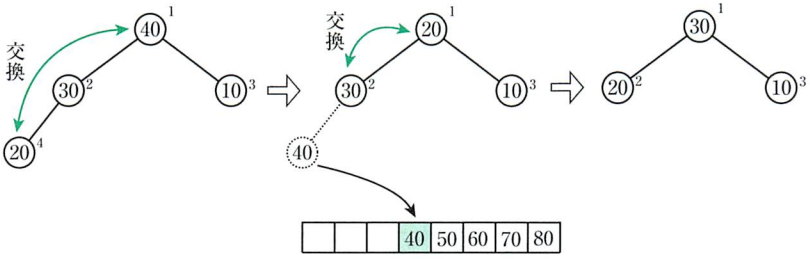
図③　60を取出す



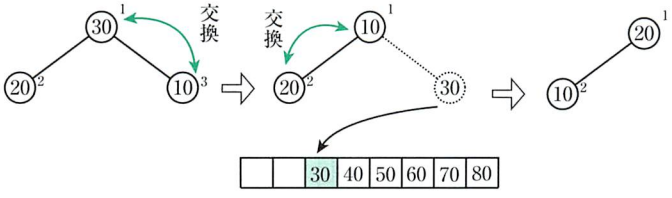
図④　50を取出す



図⑤　40を取出す



図⑥　30を取出す



2　子のうち大きいほうを親と比較して，大きければ交換する

根の最大値を末尾に設定します。図①では80を8番に設定しています。新たな根となった20が子よりも大きいかを調べます。子の番号は，1×2＝2と1×2＋1＝3なので，2番と3番のデータの大きいほうを根と比較し，大きければ入れ替えます。ここでは，70と60なので70を根とします。同様に，2番の子は2×2＝4と2×2＋1＝5なので，4番と5番のデータの大きいほうの50と比較し，大きいので20と50と入れ替えます。

224ページの流れ図と照らし合わせて，ヒープソートの仕組みを良く理解しておきましょう。