

目次

1	はじめに	2
2	バブルソート	3
3	バケットソート	4
4	ヒープソート	5
4.1	二分ヒープ	5
5	挿入ソート	7
6	マージソート	8
7	クイックソート	9
8	時間計測	10
9	おまけ	12

1 はじめに

コードは全部 **ここ**にあるので、コードは直接見てください。
考える配列の長さは特に断りがない限り、 N とする。

2 バブルソート

- 「隣り合う要素を比較して、順番が逆であれば交換する」というのを「操作」と呼ぶことにする。バブルソートは、「操作」を繰り返すことで、数列を昇順に並べ替えるアルゴリズムである。
- i 回の「操作」を行うと、数列の右端から少なくとも i 個の要素が正しい順番に並ぶ。1回の「操作」で $N - i + 1$ 回の比較を行い、ソート全体で「操作」が行われる回数は高々 N 回なので、時間計算量は $O(N^2)$ である。
- ソートが完了していたら、直ちに処理を終了するような実装をする場合、1回の「操作」で終われば、時間計算量は $\Theta(N)$ でこれが最良である。
- 逆に、完全に逆順になっている数列に対してバブルソートを行うと、 N 回の「操作」を行うことになり、時間計算量は $\Theta(N^2)$ となる。

3 バケットソート

- 最大値を R , 最小値を L とする. $R - L + 1$ の大きさの配列をもって, 各値が何回出現したかを記録して, その情報をもとにソートする.
- $M = R - L + 1$ と置けば, 時間計算量は $O(M + N)$, 空間計算量は $O(M)$ である.
- M がかなり大きいと空間がすごく無駄になるので, 座標圧縮したくなるが, 結局座標圧縮でソートしないといけないので本末転倒である.

4 ヒープソート

- 二分ヒープを用いたソートアルゴリズムである。
- 根の取得 $O(1)$ と，根の更新 $O(\log N)$ が N 回行われるので，時間計算量は $O(N \log N)$ である。

4.1 二分ヒープ

ヒープソートはin-placeにやるので，0-indexedの二分ヒープを使う．ノードの添え字は，図1のようにつけ，それぞれのノードは値を持っている．

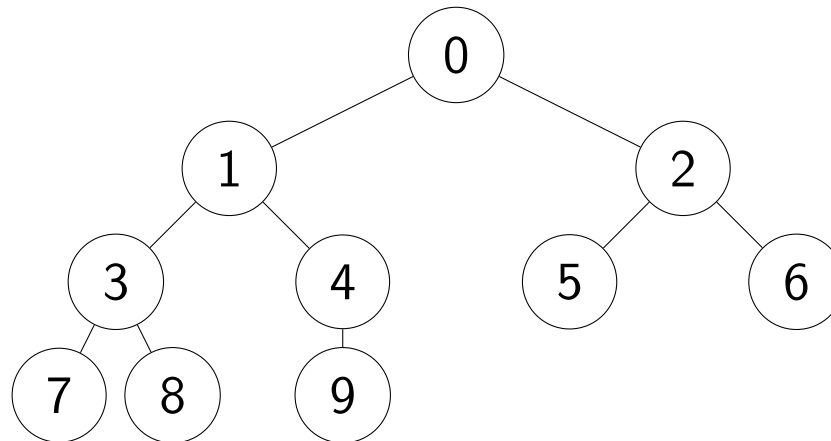


図1 二分ヒープのindex

親ノードの値は，子ノードの値以上になるようにする．構築は，葉から根に向かって，違反

があればswapするというのを繰り返せばよく，根の更新は，根から葉に向かって，違反があればswapするというのを繰り返せばよい．最初の構築は $O(N)$ ，再構築は高さが $O(\log N)$ なので $O(\log N)$ である．

5 挿入ソート

- 「 i 番目から 0 番目までこの順番で見えていて、逆の順番があれば交換する」というのを「 i 番目の操作」と呼ぶことにする．挿入ソートは、 $i = 1, \dots, N - 1$ に対して、この順番で i 番目の操作を行うことで、数列を昇順に並べ替えるアルゴリズムである．
- j 番目の操作までを行ったとき、 j 番目まではソート済みだから、 $j + 1$ 番目の操作では、正しい順番になった瞬間 $j + 1$ 番目の操作を終了することができる．そのため、すでにソートされている場合、 i 番目の操作は $O(1)$ だから、時間計算量は $O(N)$ でこれが最良である．
- 逆に、完全に逆順になっている数列に対して挿入ソートを行うと、 i 番目の操作では、 i 回の比較が行われるため、時間計算量は $\Theta(N^2)$ となる．

6 マージソート

- マージソートは，分割統治法を用いたアルゴリズムである．
- $N > 2$ 個の要素を $N/2$ 個の要素に分割し，それぞれを再帰的にソートし，2つのソート済みの配列の大きくないほうを順番に取り続けるという操作で， N 個の要素をソートする． $N \leq 2$ の時は，普通にする．
- マージソートの時間計算量を $T(N)$ とおくと， $T(N) = 2T(N/2) + \Theta(N)$ なので， $T(N) = \Theta(N \log N)$ である ($\Theta(N)$ の $\Theta(\log N)$ 個の和になっている．) ．

7 クイックソート

- クイックソートは、マージソートとだいたい同じだが分割の仕方が違う．
- ピボットと呼ばれる値を1つ選び、それ未満とそれ以上に分割する．ピボットの選び方が悪いと、1個と $N - 1$ 個の分割が繰り返されることになって、時間計算量が $\Omega(N^2)$ になる．
- ここでは、0 番目の値をピボットとして選ぶことにしているので、完全に逆順になっていたり、最初からソート済みであったりしたら、時間計算量は $\Theta(N^2)$ である．

8 時間計測

- ここでは、 $N \leq 10^6$ のときの時間計測を行う。
- 表1は、0から $N - 1$ までの整数をランダムに割り当てたときの時間である。
- 表2, 3は、最悪の場合の時間である。

表1 ランダムな数列の時間 (単位: ms)

N	バブル	バケット	ヒープ	挿入	マージ	クイック
10^3	1	1	1	1	1	1
10^4	185	1	1	7	1	1
10^5	114176	5	50	41169	68	51
10^6	—	5	65	—	82	51

表2 最悪の場合の時間 (バケット) (単位: ms)

M	バケット
10^7	61
10^8	421
10^9	4198

表3 最悪の場合の時間 (挿入, クイック) (単位: ms)

N	挿入	クイック
10^3	1	1
10^4	800	391
10^5	80812	Error

9 おまけ

一般に全順序が与えられたときでもソートができる．比較が $O(1)$ なら，同様に $O(N \log N)$ でできる．例えば，逆順にソートしたいときとか，前半に偶数，後半に奇数を入れるみたいな特殊なソートもできる．詳しくは，最初に示したリンクから見てください．