

アルゴリズムとデータ構造

演習第 9 回

ソート 2（最速のソート）

最も速いソート方法であるクイックソート（[アルゴリズム C 第 1 巻 p.133](#)）について学びます。

問題 1 [\[印刷用 PostScript\]](#)

(1) 次のように並んでいる数列を、マージソートを用いてソートしなさい。ただし、途中の過程も書くこと。

5 1 4 3 8 2 6 7

(2) 以下はクイックソートを行う関数と、それに対するコメント文である。各コメントがプログラムのどの部分に対応するかを書きなさい。

<pre>quicksort(int a[], int l, int r){ int v, i, j, t; if(r>l){ v=a[r]; i=l-1; j=r; for(; ;){ while(a[++i]<v); while(a[--j]>v); if(i>=j) break; t=a[i]; a[i]=a[j]; a[j]=t; } t=a[i]; a[i]=a[r]; a[r]=t; quicksort(a, l, i-1); quicksort(a, i+1, r); } }</pre>	<p>一番右端の要素を軸にする i と j を初期化する i と j がぶつかるまで繰り返す i を軸よりも大きい要素が 見つかるまで右へ進める j を軸よりも小さい要素が 見つかるまで左へ進める i と j がぶつかったらループを抜ける i の指す要素と j の指す要素を交換する a[i] と軸を交換する 軸より左を再びクイックソート 軸より右を再びクイックソート</p>
---	--

(3) 次のように並んでいる数列を、(2) の関数で行っている処理と同じようにして分割しなさい。

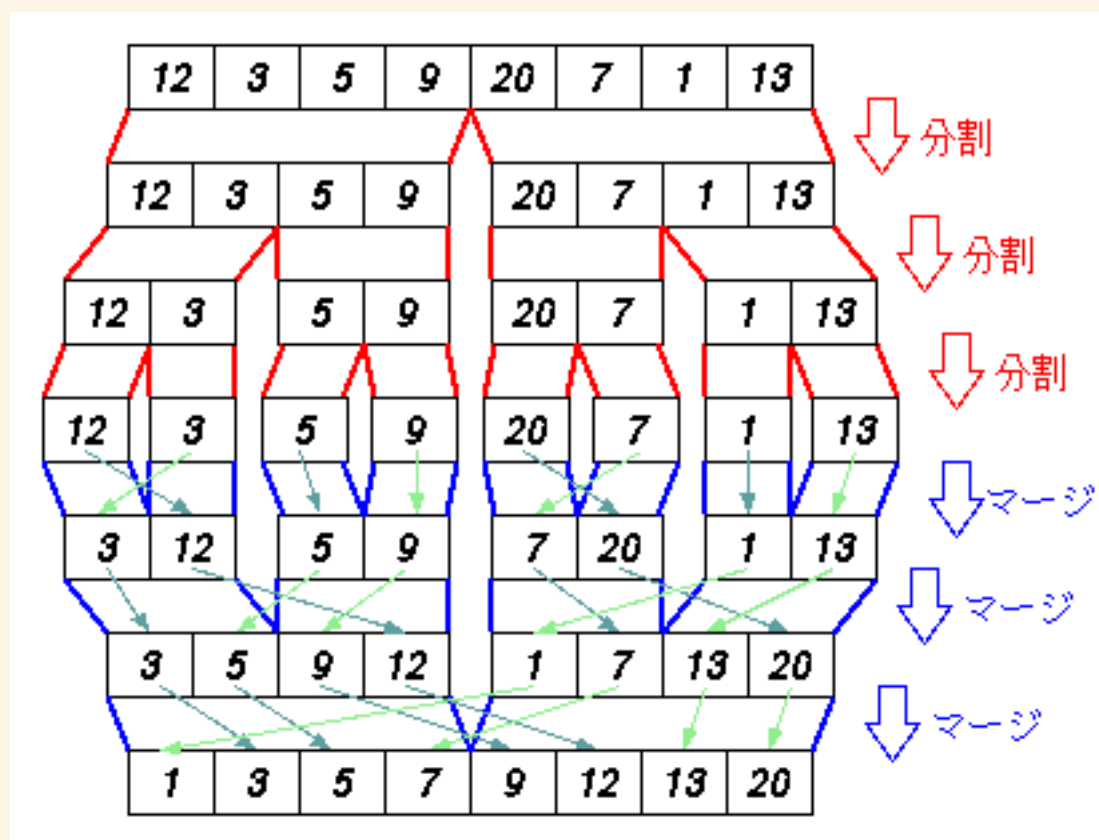
5 7 3 6 2 8 1

(1) マージソート（[アルゴリズム C 第 1 巻 p.185](#)）は、細かく分割して、それを並び直しながら再び統合（マージ）していくソート方法です。分割する分少し手間がかかって遅そうですが、ソートされているもの同士を統合することは簡単にできます。例えば次の図のようにソートされた二つの配列がある場合、先頭同士を比べて、小さい方を取って並べてゆけばよいことになります。

1	3	5	7	9	12
---	---	---	---	---	----

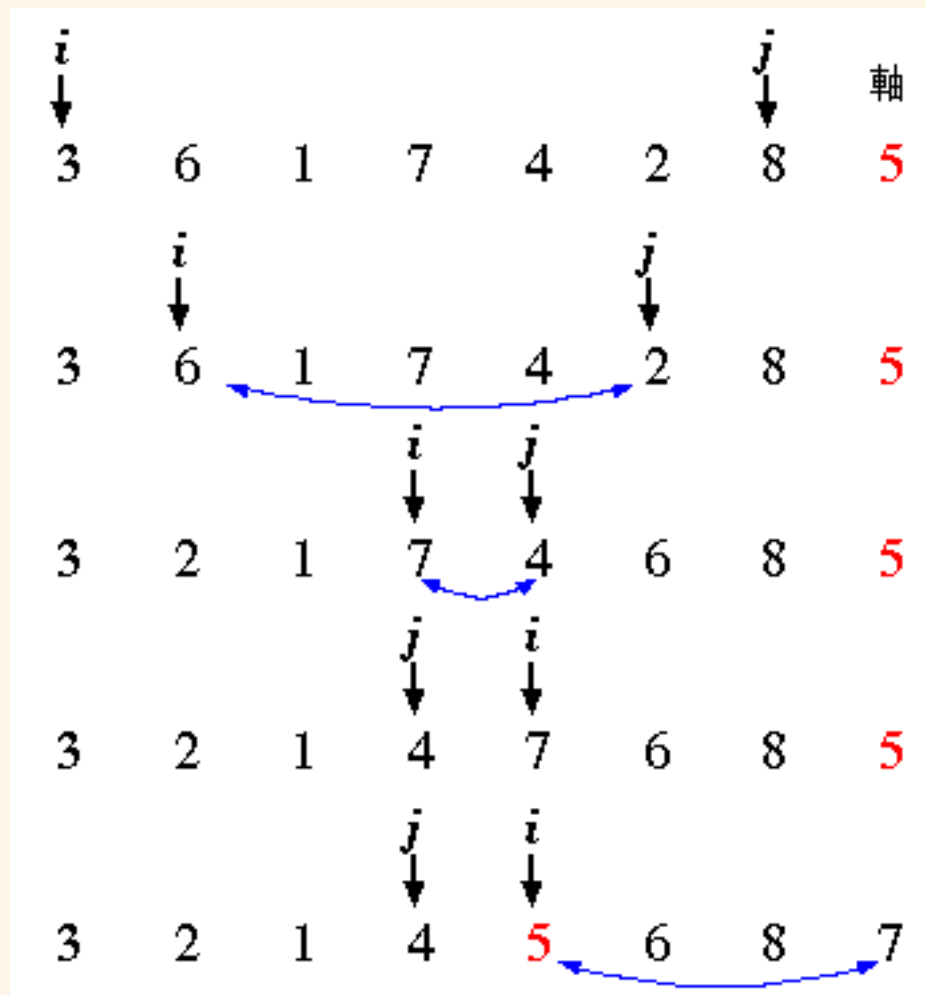
13	20
----	----

下の図を参考にして、解いてみてください。



(2) クイックソートは、軸を適当に決めて軸の値より小さい方と 大きい方に分け（分割）、分けたものに対して再び同様のことをしていく方法です。適当でもできてしまいそうですが、分割の方法を中心にプログラムをよく読んで 理解してください。

(3) 次の図を参考に、(2) での分割処理を実際の数字で確認してみてください。



クイックソートを行う関数を作りなさい。プログラムは以下の条件を満たすこと。
([ex09-2-skel.c](#))

- 使用するデータの個数は 100000 個とする
- データは乱数を用いて作成する
- ソートの途中や結果は表示しなくても良い
- 時間計測には、[演習第 5 回](#) で使用した `gethrtime` 関数を使用する

まずはデータの個数を 10 個くらいにし、配列 a の内容を ソート前とソート後に表示して、うまく動いているかを確認してください。

[演習第 8 回の問題 3](#) をやった人は、シェルソートよりもどのくらい速いか確認してみてください。

問題 3

教科書に載っている `quicksort` 関数には実は間違いがある。データがある条件を満たしている場合は、うまくソートすることができない可能性がある。どのような条件のときにうまく動かないかを考え、プログラムを修正しなさい。

教科書のままでも大抵のケースはうまく動いてしまいがちですが、Segmentation Fault になる可能性もあります。

Written by わかまつなおき