

`merge` の実現方法に気をつければ、安定なソートアルゴリズムとして実現することができる。

### 第7章のポイント

- 分割統治法とは、問題をいくつかの部分問題に分割して解いた後、その解を再構成して全体の解を得るという手法であり、分割、統治、組合せという3つのステップで構成される。
- 分割統治法における分割のステップでは、各部分問題は再帰的に解かれることが多いので、分割統治法のアルゴリズムは再帰を用いることが一般的である。
- $n$  桁の整数どうしの掛け算を基本的な方法で計算すると、 $O(n^2)$  時間が必要であるが、分割統治法を用いると、 $O(n^{\log_2 3}) = O(n^{1.59})$  時間で計算することができる。
- 分割統治法の代表例であるマージソートは、入力のデータを2つに分割し再帰的にソートを行うソートアルゴリズムである。マージソートでは、サイズが  $\frac{n}{2}$  の2つのソート済みの列を  $O(n)$  時間でマージする処理を関数 `merge` により実現することにより、 $n$  個のデータに対して  $O(n \log n)$  時間でソートを実行することができる。

### 演習問題

7.1 以下の文章の①～④について、それぞれ正しい記号を下から選べ。正しい記号が複数存在する場合はすべて列挙せよ。

分割統治法は、(①)というアルゴリズムであるが、(②)。マージソートは、(③)というソートアルゴリズムであるが、 $n$  個のデータに対しては、(④)。

- ① : a. すべての解を効率良く列挙する  
b. アルゴリズムの実行途中において全体的なことは考えず、局所的に最良の解を選択する  
c. 入力をいくつかの部分問題に分割し、各部分問題を再帰的に解く  
d. 問題を部分問題から解き、その解を記録しておいて再利用する
- ② : a. 再帰とともに用いられることはない  
b. 分割、統治、組合せという3つのステップで構成される  
c. 入力は必ず2つの部分問題に分割される  
d. 時間計算量は、入力サイズ  $n$  とすると必ず  $O(n \log n)$  となる
- ③ : a. 配列の左からデータを順番に処理する  
b. データをほぼ同じサイズの2つの集合に再帰的に分割する  
c. 入力を基準値を用いて2つの集合に分割し、再帰的にソートを実行する  
d. 配列以外のデータ構造を利用する
- ④ : a. 時間計算量がつねに  $O(n \log n)$  である  
b. 最悪時間計算量はクイックソートの最悪時間計算量と同じである  
c. つねに挿入ソートより高速に実行できる

- d. 入力により時間計算量が異なる

7.2 アルゴリズム 7.3 のマージソートを改良し、以下のように  $3$  つの  $\frac{n}{3}$  個のデータに分割するアルゴリズムを考えた。

```
mergesort2(D,left,right) {
    p1=(left+right)*(1/3); p2=(left+right)*(2/3);
    if (left<p1) { mergesort2(D,left,p1); }
    if (p1+1<p2) { mergesort2(D,p1+1,p2); }
    if (p2+1<right) { mergesort2(D,p2+1,right); }
    merge2(D,left,p1,p2,right);
}
```

ここで、関数 `merge2` は、サイズが  $\frac{n}{3}$  の3つのソート列を  $O(n)$  時間でマージすることができる関数だとする。

- (1) アルゴリズム全体の時間計算量  $T(n)$  は、 $T(n) = a \times T\left(\frac{n}{b}\right) + O(n)$  と表すことができる。 $a$  と  $b$  の値を答えよ。
- (2) (1)の式をもとに、このアルゴリズムの動作を表す再帰木を描け。
- (3) (2)の再帰木を用いて  $T(n)$  を求めよ。

7.3 以下の問いに答えよ。

- (1) 9枚のコインの中に1枚だけ重さの軽い偽コインがある。天秤を2回だけ使って、この偽コインをみつける方法を文章で説明せよ。
- (2) (1)のアイデアを使って、 $n$ 枚のコインの中から1枚だけ重さの軽い偽コインをみつけるアルゴリズムを作成せよ。
- (3) (2)のアルゴリズムの時間計算量を理由とともに示せ。