

## 第二回全統かんだ模試

神田幸大

July 15, 2016

### 1 アルゴリズム

#### 1.1 アルゴリズムの設計方法について、以下の文章の空欄に適切なものを、下の語群から選び答えよ。

ある制約に基いて目的関数の最小値または最大値を与えるような解を求める問題を (A) という。この問題を解く手法として、問題を部分問題に分割し、各部分問題の最適解を保存して再利用する (B) と、ある基準に基いて、各段階での最良解を連続して選択する (C) がある。それぞれの手法およびそれにより解決できる問題の具体例として、(B) は (D)、(C) は (E) が挙げられる。

(語群)

ダイクストラ法、ボイヤー・ムーア法、ナップサック問題、最適化問題、再帰、動的計画法、分割統治法、貪欲法

#### 1.2 硬貨問題について以下の問に答えよ。

硬貨問題とは、与えられた硬貨の種類  $a_1, a_2, \dots, a_n (i \neq j \Rightarrow a_i \neq a_j)$  について、金額の和が目的金額  $N$  になるような硬貨の選び方のパターンのうち、使用枚数の合計が最も少なくなるような選び方を求める問題である。このとき、各硬貨の使用枚数に上限および下限は無いとする（使わない硬貨があってもよい）。

##### 1.2.1 以下のそれぞれの場合について解 (合計枚数およびその内訳) を答えよ。

ア  $a[4] = \{10, 50, 100, 500\}, N = 770$

イ  $a[4] = \{10, 40, 90, 150\}, N = 220$

- 1.2.2 1.2.1 のアおよびイの解は、計算量の観点から、それぞれどのような手法を用いるのが適当か。1.1 の語群から選んで答えよ。
- 1.2.3 1.2.2 のように、硬貨問題では、硬貨の種類および目的金額によって解を求める手法が異なる場合がある。1.2.2 のイについて、「アについて適当な手法」を用いたときに得られる解を答えよ。このとき、イの最適解と異なっているようにすること。

## 2 データ構造

- 2.1 データ構造について、以下の文章の空欄に適切なものを、(B) 以外に関しては下の語群から選び答えよ。(B) に関しては適切な文で埋めよ。

二分探索法に対応したデータ構造として、木構造を採用した (A) がある。これは、各節点  $v$  について、 $v$  の左部分木のデータは全て  $v$  以下であり、 $v$  の右部分木のデータは全て  $v$  以上であるような条件を満たす。この構造を用いて二分探索を行う場合、構造によっては効率が悪い可能性がある。例えば、一連のデータ列を挿入する際に、(B) であるようなデータ列の挿入を行うと、探索の際の平均探索時間が (C) となってしまう、そうでない場合に比べて効率が悪い。

そこで、ある制約に基いて構造を変形させるようにしたものを特に (D) と呼ぶ。(D) の具体例として (E) や (F) などが挙げられる。(E) は、(F) に比べて制約が厳しいため、データの挿入・削除の度に、変形による計算負荷が大きくなる可能性がある。すなわち、データの挿入・削除が少ない場合は (E) の構造が適しており、そうでない場合は (F) の構造が適している。

(語群)

ヒープ、二分木、スタック、 $O(n^2)$ 、 $O(\log n)$ 、 $O(n)$ 、AVL 木、平衡二分探索木、赤黒木 (二色木)

- 2.2 任意の節点に関して、その左部分木の高さと右部分木の高さの差が高々1 であるような木構造を考える。以下の問に答えよ。
- 2.2.1 この木構造の名称を 2.1 の語群から選んで答えよ。
- 2.2.2 この木構造において、データ挿入の平均計算量をオーダー表記で答えよ。
- 2.2.3 この木構造の制約を満たすための変形方法として、左回転および右回転がある。あるデータの挿入によって次の図のような構造になったとき、制約を満たすまでの過程を図とともに示せ。左回転・右回転のどちらを用いたかも明記すること。

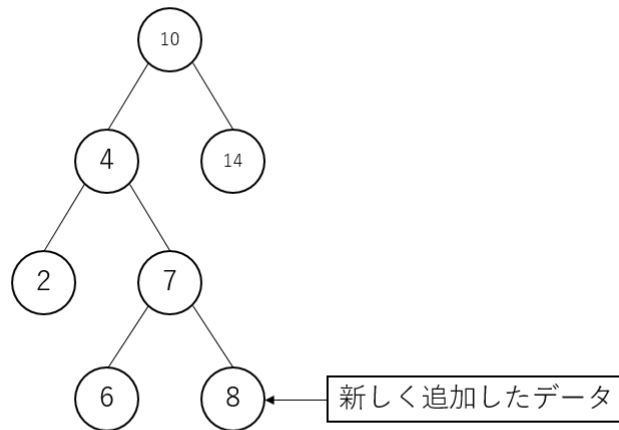


Figure 1: 問 2.2.3

### 3 ソート

#### 3.1 次に示すプログラムについて以下の問に答えよ。

次に示す C 言語のプログラムは、配列 `data` の要素に格納された非負整数のデータを、関数 `sort` を再帰させることによって昇順に整列し出力するプログラムである。

- 3.1.1 プログラムで実現されている整列アルゴリズムは一般になんと呼ばれているか名称を答えよ。
- 3.1.2 空欄 9 行目の (ア)、26 行目の (イ)、34 行目の (ウ) を適切に埋めよ。
- 3.1.3 15 行目および 19 行目の `for` 文では、配列 `x` のデータを配列 `tmp` にコピーしている。このとき、配列 `x` のデータ順を保ったままコピーしている部分と、逆順にコピーしている部分がある。この手法の利点を述べよ。ただし、このようにしなかったときのプログラムの実装方法と比較するようにせよ。
- 3.1.4 データの個数が  $n$  個のとき、この整列アルゴリズムの平均計算量をオーダー表記で答えよ。 $n = 2^k, k \in \mathbb{N}$  と仮定してよい。

```

1  #include <stdio.h>
2  #define MAX 10
3
4  int tmp[MAX];
5
6  void sort(int x[],int l,int r){
7      int m,i,j,k;
8
9      if((l>=r)) return;
10
11     m=(l+r)/2;
12     sort(x,l,m);
13     sort(x,m+1,r);
14
15     for(i=l;i<=m;i++){
16         tmp[i]=x[i];
17     }
18
19     for(i=m+1,j=r;i<=r;i++,j--){
20         tmp[i]=x[j];
21     }
22
23     i=l;j=r;
24
25     for(k=l;k<=r;k++){
26         if((k%2)) x[k]=tmp[i++];
27         else x[k]=tmp[j--];
28     }
29 }
30
31 void main(){
32     int i;
33     int x[]={9,4,6,2,1,8,0,3,7,5};
34     sort((x));
35
36     for(i=0;i<MAX;i++){
37         printf("%d\t",x[i]);
38     }
39 }

```