

2006 年度ソフトウェア工学試験問題

日時：平成 18 年 7 月 28 日(金)12:50-14:20 場所：K205

注意事項：

- ① 指定された席に着席すること。
- ② 答案は机に残して退室すること。
- ③ 問題用紙は持ち帰ること。
- ④ 試験用紙、計算用紙が必要なときには申し出ること。
- ⑤ 解の簡単な導出過程も書くこと。

1. 漸近量評価

(1) 次の関数を O 記法で示せ。(出来るだけ漸近計算量が小さいもので表すこと。)

(a) $T_a(n) = 0.5n^2 - 10n + 5$

(b) $T_b(n) = 5(n+1)(n-2)(n+3) + 7(n+5)(n-3) + 32$

(c) $T_c(n) = \sqrt{n} + \log n$

(d) $T_d(n) = n + \log n + 0.5^n$

(e) $T_e(n) = n^5 + 2^{\frac{n}{5}}$

(2) 次の擬似コードの漸近計算量を O 記法で示せ。

ただし、以下のコードでは引数 n が入力サイズとし、すべて $n = 2^k$ の形であるとする。

(a) 漸近計算量 $T_A(n)$

アルゴリズム A:

```
void algoA(int n){
    for(i=0; i<n; i++){
        for(j=i; j<n; j++){
            (定数  $c_1$  時間の処理)
        }
    }
    return;
}
```

(b) 漸近計算量 $T_B(n)$

アルゴリズム B:

```
void algoB(int n){
    for(i=2; i<n; i=i*i){
        (定数  $c_2$  時間の処理)
    }
    return;
}
```

(c) 漸近計算量 $T_c(n)$

アルゴリズム C :

```
void algoC(int n){
    if(n<=0){
        return;
    }else{
        algoC(n-1);
        (定数  $c_3$  時間の処理)
        algoC(n-1);
        return;
    }
}
```

2. データ構造

2 分探索木 T に集合 $S = \{1, 3, 4, 5, 7, 8, 10\}$ を蓄えることを考える。2 分探索木 T にデータ x を挿入する操作を **insert(x)** で表し、2 分探索木 T からデータ x を削除する操作を **delete(x)** で表す。このとき、以下の問いに答えよ。

(1) 集合 S を蓄える 2 分探索木で高さが最大のものを 1 つ図示せよ。(1 種類示せばよい。)

(2) 集合 S を蓄える 2 分探索木で高さが最小のものを 1 つ図示せよ。(1 種類示せばよい。)

(3) 空の (要素数 0 の) 2 分探索木 T に次の順序で集合の要素を挿入するとき、2 分探索木の状態の変化を図示せよ。

insert(7) → insert(3) → insert(8) → insert(5) → insert(4) → insert(1) → insert(10)

(4) (3) の木から次の操作を行ったときの 2 分探索木を変化を図示せよ。ここで、**delete(x)** はデータ x を削除する操作であり、**insert(x)** はデータ x を挿入する操作である。
delete(3) → insert(9) → delete(5) → delete(4) → delete(8)

(5) 空の 2 分探索木からある順序で挿入を繰り返して、(2) の木が得られた。このような挿入順序を 1 つ示せ。

3. ヒープソート

配列 A には n 個の要素 $a_0, a_1, \dots, a_{n-1}$ が $A[0], A[1], \dots, A[n-1]$ にそれぞれ蓄えられているとする。この（単一の）配列 A だけを用いてヒープソートを行うアルゴリズムを下に擬似コードで示す。ここで、ヒープは親の方が大きい要素を蓄えるものとする。（すなわち、ある頂点 v を根とする部分木中の最大の値を頂点 v に蓄える。）また、`print_array(n, A)` は配列 A の最初の n 要素の内容を表示する関数である。このアルゴリズムについて問いに答えよ。

ヒープソート：

```
void h_sort(int n, int A[]){
    /*ヒープ状態作成*/
    for(i=0; i<n; i++){
        (A[i]をヒープに挿入し、A[0]からA[i]までをヒープ状態にする。)
        print_array(n, A);
    }
    /*並べ替え*/
    for(i=n-1; i>0; i--){
        (A[i]とA[0]を交換し、A[i]からA[n-1]までをソート状態にする。)
        print_array(n, A);
    }
    return;
}
```

(1) $n=8$ とし、配列 A に下のように値が蓄えられているとする。`h_sort(8, A[])` を実行したときに `print_array(n, A)` の各命令により表示される配列の内容を示せ。また、`print_array(n, A)` の各命令が実行されたときのヒープの状態を木として図示せよ。

	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]
A	18	46	7	83	21	75	39	55

(2) 一般に、高さが h のヒープに蓄えることのできる最大のデータ数 N を求めよ。（なお、根の高さは 0 として扱うこと。）

(3) ヒープに n 個のデータが蓄えられているとする。このとき、ヒープの高さ h とデータ数 n との関係を式で表せ。また、高さ h をデータ数 n を用いた O 記法で示せ。

(4) ヒープソートの時間計算量を O 記法で示せ。ただし、`print_array(n, A)` に用いられる時間計算量を含めないこと。

4. ハッシュ

アルファベット $A = \{a, b, \dots, z\}$ 中の 3 文字からなる 5 個の名前 $N = \{abe, ebi, kan, ito, oki\}$ を要素数 10 の配列 X にハッシュ法で蓄えることを考える。

ハッシュ関数 $h: A^3 \rightarrow \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ は以下の式を用いるものとする。

$\mathbf{x} = x_1 x_2 x_3 \in A^3$ に対して、

$$h(\mathbf{x}) \equiv \sum_{i=1}^3 c(x_i) \pmod{10}$$

ここで、 $c: A \rightarrow \mathbb{Z}$ は以下に示すようなアスキーコードである。($\mathbb{Z}$ は整数の集合。)

$$c(a) = 97, c(b) = 98, \dots, c(z) = 122$$

また、要素 $\mathbf{x}$ を挿入する際に k 回衝突が起きたとすると、次のハッシュ関数値で得られた値を添え字として配列 X に蓄えるとする。

$$h_k(\mathbf{x}) \equiv h(\mathbf{x}) + k \pmod{10}$$

このとき、次の問いに答えよ。

(1) 各要素 $\mathbf{x} \in N$ に対して、ハッシュ値 $h(\mathbf{x})$ を求めよ。

(2) 空の配列へ順に N の要素をすべて挿入するときの衝突回数の総和 $K = \sum_{\mathbf{x}_i \in N} k_i$ を求めよ。

ここで、 k_i は要素 x_i を挿入するときの衝突回数である。

(3) 次の順序で名前を空の配列に挿入するとき、配列 X の内容の変化を図示せよ。

abe $\rightarrow$ ebi $\rightarrow$ kan $\rightarrow$ ito $\rightarrow$ oki

(4) ある順序で名前を空の配列に挿入したら、次の配列が得られた。このような格納状態になる挿入順序を全て示せ。

	X[0]	X[1]	X[2]	X[3]	X[4]	X[5]	X[6]	X[7]	X[8]	X[9]
X			ito	oki	dai	kan	ebi	abe		