

2008 年度ソフトウェア工学㊦試験問題

日時：2008 年 9 月 5 日(金)12:50-14:20 場所：K205

注意事項：

1. 答案は机は教卓に提出すること。
2. 問題用紙は持ち帰ること。
3. 試験用紙、計算用紙が必要なときには申し出ること。
4. 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) $T_a(n) = 2n^5 - 9n^3 - 7n$

(b) $T_b(n) = 2(\sqrt[3]{n}+1)(\sqrt[3]{n}-2) + (\sqrt{n}+5)(\sqrt{n}-3) + 3(n+5)(n-5)$

(c) $T_c(n) = (\log n + \sqrt{n})^5 + (2^{\log n} + \sqrt{n})^3$

(d) $T_d(n) = 4^{\log n} + \sqrt{n^3} + (\log n)^5$

(e) $T_e(n) = \frac{3(\sqrt{n}-8)(\sqrt[3]{n^2}-2)}{\sqrt{n^3}+5} + \frac{2(\sqrt[5]{n^3}-4)(\sqrt{n^2}-7)}{\sqrt[3]{n^5}+3}$

(2) 次の C 言語風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=0; j<i; j++){
            (定数  $c_a$  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=1; i<n; i=2*i){
        for(j=0; j<i; j++){
            (定数  $c_b$  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<=1){
        return;
    }else{
        (定数  $c_c$  時間の処理)
        algo(n/2);
        return;
    }
}
```

(d) 漸近計算量 $T_d(n)$

アルゴリズム D :

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

2. データ構造 (ヒープ)

配列 A を利用してデータ構造のヒープを作成する。ヒープは次の条件を満たすように構築されているとする。

- $A[0]$ を根 r とし、根 $A[0]$ には最小値が蓄えられる。
- 根以外の各頂点 v に対して、親の方が小さい要素を蓄えているとする。(すなわち、頂点 v を根とする部分木中の最小の値が頂点 v に蓄えられる。)
- 頂点 v が $A[i]$ に蓄えられているとき、頂点 v の左の子は $A[2i+1]$ に、右の子は $A[2i+2]$ に蓄えられるとする。

このとき、次の設問に答えよ。

(1) 配列が次の状態であるとき、ヒープの木の形状を図示せよ。

	$A[0]$	$A[1]$	$A[2]$	$A[3]$	$A[4]$	$A[5]$	$A[6]$	$A[7]$	$A[8]$	$A[9]$	$A[10]$	$A[11]$
A	12	15	21	52	38	27	41	73				

(2) (1) の状態から次のように操作を行うとする。このとき、配列の状態とヒープの木をそれぞれ示せ。

$get_min() \rightarrow insert(45) \rightarrow get_min() \rightarrow get_min()$

$\rightarrow insert(8) \rightarrow insert(19) \rightarrow get_min() \rightarrow insert(30)$

(3) 一般に n 個のデータが蓄えられているときに、操作 1 回に必要な最悪時間計算量を求 O 記法で示せ。ただし、結果だけでなく、理由も記述すること。

3. 選択ソート

配列 B に n 個のデータが蓄えられている。このとき、最大値を求めるアルゴリズムを利用した選択ソートにより、配列 B を昇順にソートしよう。このようなアルゴリズムを下に示す。

```
int find_max(int left,int right){
    int j,max=left;
    for(j=left+1;j<=right;j++){
        if(B[max]<B[j]){
            max=j;
        }
    }
    return max;
}
void select_sort(){
    int i,max;
    for(i=n-1;i>0;i--){
        max=find_max( (ア) );
        swap( (イ) );
    }
    return;
}
```

ここで、`find_max` は配列 B の $B[\text{left}] - B[\text{right}]$ 中の最大値を保持している配列要素 $B[j]$ の添え字 j を返す。

- (1) 選択ソートが適切に動作するように、(ア)、(イ) を適切に埋めよ。
- (2) 1 回の `find_max` 呼び出しに必要な最悪時間計算量 $T_{\max}(n)$ を示せ。
- (3) アルゴリズム全体における交換回数 $N_{\text{swap}}(n)$ を示せ。なお、 $N_{\text{swap}}(n)$ は O 記法ではなく、厳密な式で答えること。
- (4) アルゴリズム全体の最悪時間計算量 $T_{\text{sort}}(n)$ を O 記法で示せ。

4. 2分探索

配列 C に次のようなデータが蓄えられているとする。このとき、下に示すプログラムで2分探索を行おうとしている。

	$C[0]$	$C[1]$	$C[2]$	$C[3]$	$C[4]$	$C[5]$	$C[6]$	$C[7]$
C	2	11	21	28	35	52	88	91

```
int binary_search(int key,int left,int right){
    int mid;/*中央の添え字*/
    if(left>right){return -1;/*存在しない*/}
    else{
        mid=(left+right)/2;
        if(C[mid]==key){
            return mid;
        }else if(key < C[mid]){
            return binary_search( (ア) );/*小さい方*/
        }else{
            return binary_search( (イ) );/*大きい方*/
        }
    }
}
```

- (1) 2分探索が適切に動作するよういに、(ア)、(イ)に入る適切な引数を示せ。
- (2) 次の引数でこの関数を呼び出したとき、調べられる配列の要素を順に示せ。
(a) `binary_search(21,0,7);`
(b) `binary_search(65,0,7);`
- (3) データ数が一般の n であるとき、上のプログラムの時間計算量 $T(n)$ が満たすべき漸化式を示し、最悪時間計算量 $T(n)$ を O 記法で答えよ。