H17 年度ソフトウェア工学試験問題解答例

１．漸化式の計算（２０点）

フィボナッチ数列は、次の漸化式で定義される。

$$f(n)=\begin{cases}0 & n=0\\ 1 & n=1\\ f(n-1)+f(n-2) & n\geq 2\end{cases}$$

このフィボナッチ数列を計算するアルゴリズム２つを、C 言語風に以下に示す。なお、変数宣言等は省略している。

アルゴリズム A:

```
int   algoA(int n){
        if(n==0){
                return 0;
        }else if(n==1){
                return 1;
        }else{
                return algoA(n-1)+algoA(n-2);
        }
}
```

アルゴリズム B :

```
int   algoB(int n){
        F[0]=0;
        F[1]=1;
        for(i=2;i<=n;i++){
                F[i]=F[i-1]+F[i-2];
        }
        return F[n];
}
```

この２つのアルゴリズムに関して、以下の問いに答えよ。

（１） 入力サイズをnとしたとき、 アルゴリズム A の最悪時間計算量$T_A(n)$が満たすべき漸化式を示せ。（５点）

c_0, c_1, c_2 を定数として、次式のように表現できる。

$$T_A(n) \le \begin{cases} c_0 & n = 0 \\ c_1 & n = 1 \\ T_A(n-1) + T_A(n-2) + c_2 & n \ge 2 \end{cases}$$

（２）　（１）の漸化式を解き、アルゴリズム A の最悪時間計算量を O 記法で答えよ。（１０点）

時間関数の単調性より、$T_A(n-2) \le T_A(n-1)$ が成り立つ。また、$c = \max\{c_0, c_1, c_2\}$ とする。

$$T_A(n) \le T_A(n-1) + T_A(n-2) + c_2$$
$$\le 2T_A(n-1) + c_2$$
$$\le 2\left(2T_A(n-2) + c_2\right) + c_2 = 2^2 T_A(n-2) + 2c_2 + c_2$$
$$\le 2^2\left(2T_A(n-3) + c_2\right) + 2c_2 + c_2 = 2^3 T_A(n-3) + 2^2 c_2 + 2c_2 + c_2$$
$$\cdots$$
$$\le 2^{n-1} T_A(1) + 2^{n-1} c_2$$
$$= 2^{n-1}(c_1 + c_2)$$
$$\le 2^n c$$

よって、$T_A(n) = O(2^n)$ である。

（３）　アルゴリズム B の最悪時間計算量 $T_B(n)$ を O 記法で答えよ。（５点）

for ループによって、$n-1$ 回の繰り返しが起きるだけである。したがって、

$T_B(n) = O(n)$

２．クイックソート（３０点）

配列 A に次のようにデータが蓄えられているとする。

	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]
A	12	4	5	11	8	2	15	7

この配列をソートするために以下に示すようなクイックソートを用いる。

```
void   qsort(int left, int right){
         int   pos ; /*分割位置*/
         if(left>=right){
                  return;
         }else{
                  pos=partition(left,right);
                  print_array(left,right);
                  qsort(left,pos-1);
                  qsort(pos+1,right);
         }
}
```

ここで、partition はピボット（基準値）の位置を求める関数で、部分配列 A[left]-A[right] をピボットより小さい要素、ピボット、ピボットより大きい要素の順に並べかえて、ピボットの入る配列の添え字を返す関数である。また、print_array(left,right)は、部分配列 A[left]-A[right]を表示する関数である。

このとき、クイックソートにおける動作について答えよ。

（1） ピボットを部分配列の右端、すなわち、A[right]とする。このとき、print_array によって表示される配列の内容をすべて示せ。（１０点）

再帰の深さに応じて表示されることに注意する。

深さ１での表示

2	4	5	7	8	12	15	11

深さ２での表示

2	4	5

、

8	11	15	12

深さ３での表示

2	4

、

12	15

ピボットに選ばれたものは次の深さでは表示されない。また、部分配列の長さが１以下のときにも表示されないことに注意する。

（2） クイックソートが最も悪い動作をする入力例を示せ。ただし、ピボットは部分列右端をいつも選ぶものとし、データ個数は８個としてよい。（１０点）

ソート済みの列や、逆順にソートされているときが最悪である。例えば、次のような配列が最悪である。

1	2	3	4	5	6	7	8

（３） n 個のデータをソートするとき、クイックソートの最悪時間計算量をO記法で示せ
また、なぜそのような最悪時間計算量になるのかの理由も記述せよ。（１０点）

関数 partition は、部分配列の長さの時間計算量が必要である。すなわし、部分配列の長さを l とすると、$O(l)$ 時間かかってしまう。分割が、いつも１個のピボット、残りの $l-1$ のように分割されるとすると、クイックソートの最悪時間計算量 $T(l)$ は次の漸化式を満たす。（c_1, c_2 はある定数。）

$$T(l) = \begin{cases} c_1 & l = 1 \\ T(l-1) + c_2 l & l > 1 \end{cases}$$

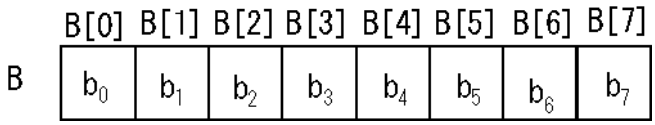
初期配列の長さは n なので、$l = n$ として漸化式を解く。

$$T(n) = T(n-1) + c_2 n$$
$$= T(n-2) + c_2 (n-1) + c_2 n$$
$$= \cdots$$
$$= T(1) + c_2 \{2 + \cdots + (n-1) + n\}$$
$$= c_2 \frac{(n+2)(n-1)}{2} + c_1$$

よって、$T(n) = O(n^2)$ のアルゴリズムである。

３．２分探索

配列Ｂに次のようにデータが蓄えられているとする。



この配列に次のようなプログラムで２分探索を行なおうとしている。

```
int  binary_search(int key,int left,int right){
        int mid;/*中間位置*/
        if(left>right){return -1;/*存在しない*/}
        else{
                mid=(left+right)/2;
                if(B[mid]==key)return mid;
                else if(key < B[mid]) {
                        return binary_search(    (a)    );/*小さい方*/
                }
                else {
                        return binary_search(    (b)    );/*大きい方*/
                }
        }
}
```

（１）２分探索に必要な条件を述べよ。（１０点）

配列の中身が昇順にソートされている。

すなわち、

$b_0 \le b_1 \le b_2 \le b_3 \le b_4 \le b_5 \le b_6 \le b_7$

が必要。

（２）(a)、（ｂ）に入る適切な記号を示せ。（１０点、各５点）

(a)：(key,left,mid-1)

(b)：(key,mid+1,right)

（３）２分探索の最悪時間計算量をＯ記法で示せ。また、なぜそのような最悪時間計算量になるのかの理由も記述せよ。（１０点）

（解法１）漸化式を立てて、解く方法。

時間計算量を$T(n)$とする。

このとき、次の漸化式を満たす。(c_1, c_2はある定数。)

$$T(n) \le \begin{cases} c_1 & n \le 1 \\ T(\frac{n}{2}) + c_2 & n \ge 2 \end{cases}$$

ここで、まず$n = 2^k$と表せるとする。
このとき、

$$T'(k) \le \begin{cases} c_1 & k = 0 \\ T'(k-1) + c_2 & k \ge 1 \end{cases}$$

と表せる。この漸化式を解く。

$$T'(k) \le T'(k-1) + c_2$$
$$\le \left(T'(k-2) + c_2\right) + c_2 = T'(k-2) + 2c_2$$
$$\le \left(T'(k-3) + c_2\right) + 2c_2 = T'(k-3) + 3c_2$$
$$\vdots$$
$$\le T'(1) + (k-1)c_2$$
$$= c_1 + (k-1)c_2$$

よって、$T'(k) = O(k)$
また、$k = O(\log n)$より、$T(n) = O(\log n)$

次に、一般のnについて考察する。$2^{l-1} \le n < 2^l$なるlが存在する。このとき、上述のように、次式が成り立つ。

$$T(n) < T(2^l) = O(l)$$
$$\therefore T(n) \le cl$$

一方、

$$l - 1 \le \log n < l$$
$$\therefore \log n < l \le \log n + 1$$

に注意すると次のように計算できる。

$$T(n) \le cl$$
$$\le c(\log n + 1) = c \log n + c$$
$$\therefore T(n) = O(\log n)$$

以上より、２分探索のアルゴリズムの最悪時間計算量は、
$O(\log n)$である。

（解法２）各ステップで探索できる要素を考えて、配列すべての要素を網羅する方法。
（配布資料では、この方法が記述されている。）
深さjの再帰までで探索が可能な要素数の最大値は、次式で表される。

$$\sum_{i=0}^{j} 2^i = 1+2+4+\cdots+2^j = \frac{2(2^j-1)}{2-1} = 2^{j+1}-1$$

最悪の場合、深さ $j-1$ までの再帰でキーが発見されず、深さ j でキーが発見される。また、全ての要素を探索する必要がある。したがって、再帰の深さの最大値に関して次式が成り立つ。

$$2^j - 1 < n \leq 2^{j+1} - 1$$

$$\therefore j < \log_2(n+1) \leq j+1$$

$$\therefore j+1 = \lceil \log_2(n+1) \rceil$$

$$\therefore j = O(\log_2 n)$$

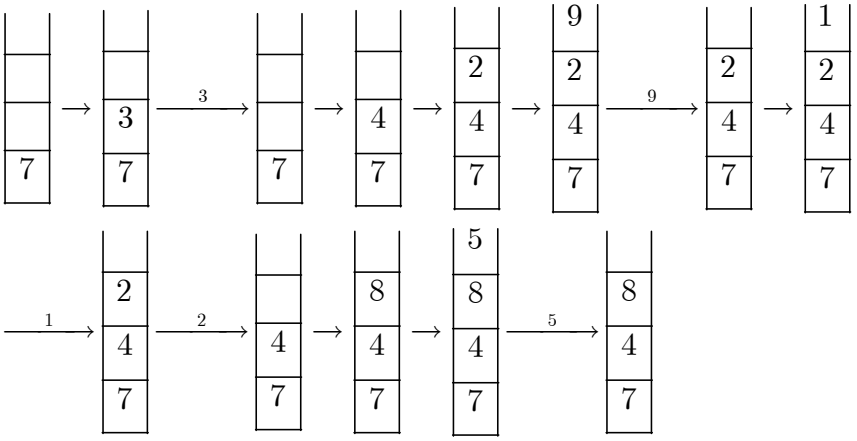
４．データ構造（２０点、各１０点）

スタックおよびキューに関して、以下の問いに答えよ。

（１）空のスタックに、次のような操作系列を行なったとき、pop()で取り出されるデータの系列および、最終的なスタックの状態を示せ。

push(7)→push(3)→pop()→push(4)→push(2)→push(9)→pop()→push(1)→pop()→pop()→push(8)→push(5)→pop()

配列の状態を逐一記述すればよい。



よって、出力系列は

$3 \rightarrow 9 \rightarrow 1 \rightarrow 2 \rightarrow 5$

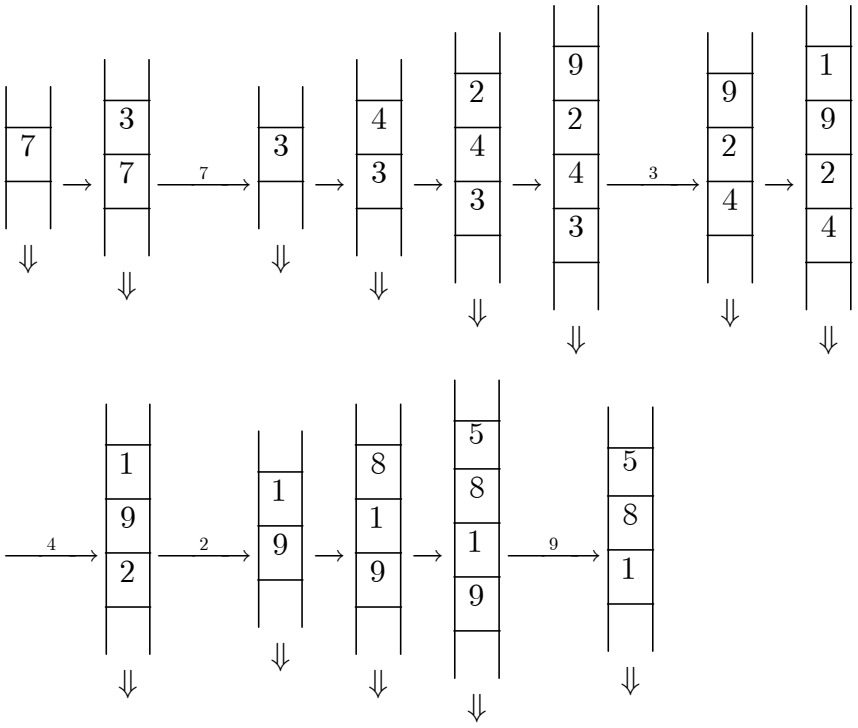
であり、最終状態は、

である。

（2）空のキューに次のような操作系列を行なったとき、dequeue（）で取り出されるデータ系列および、最終的なキューの状態を示せ。

enqueue(7)→enqueue(3)→dequeue()→enqueue(4)→enqueue(2)→enqueue(9)→dequeue()→enqueue(1)→dequeue()→dequeue()→enqueue(8)→enqueue(5)→dequeue()

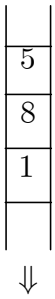
配列の状態を逐一記述すればよい。



よって、出力系列は、

$7 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 9$

であり、最終状態は、



である。