

後期中間試験問題 (2E 情報工学概論)

電気情報工学科

学籍番号

氏名

1 ソート

[問 1] 10 点

バブルソートは隣同士を比較する方法である。そして、比較が 1 回も生じないと、ソートは完了する。以下がバブルソートで、比較する値を矢印で示している。

比較の対象	比較後の交換	flag
バブルソート開始		
1巡 ↓ ↓ 2 4 1 0 3	2 4 1 0 3	0
↓ ↓ 2 4 1 0 3	2 1 4 0 3	交換発生 1
↓ ↓ 2 1 4 0 3	2 1 0 4 3	1
↓ ↓ 2 1 0 4 3	2 1 0 3 4	1
flagが1なので2巡目開始		
2巡 ↓ ↓ 2 1 0 3 4	1 2 0 3 4	交換発生 1
↓ ↓ 1 2 0 3 4	1 0 2 3 4	1
↓ ↓ 1 0 2 3 4	1 0 2 3 4	1
↓ ↓ 1 0 2 3 4	1 0 2 3 4	1
flagが1なので3巡目開始		
3巡 ↓ ↓ 1 0 2 3 4	0 1 2 3 4	交換発生 1
↓ ↓ 0 1 2 3 4	0 1 2 3 4	1
↓ ↓ 0 1 2 3 4	0 1 2 3 4	1
↓ ↓ 0 1 2 3 4	0 1 2 3 4	1
flagが1なので4巡目開始		
4巡 ↓ ↓ 0 1 2 3 4	0 1 2 3 4	0
↓ ↓ 0 1 2 3 4	0 1 2 3 4	0
↓ ↓ 0 1 2 3 4	0 1 2 3 4	0
↓ ↓ 0 1 2 3 4	0 1 2 3 4	0
flagが0なのでバブルソート完了		

[問 2] 10 点

クイックソートは基準を設けて、それより大きいグループと小さいグループに分けることをグループの数が 1 になるまで続ける方法である。以下がクイックソートで基準を矢印で示している。

2つに分ける基準	基準との大小により並び替え	基準の移動
↓ 4 5 1 3 2 6 0	4 0 1 3 2 6 5	2 0 1 3 4 6 5
↓ ↓ 2 0 1 3 4 6 5	2 0 1 3 4 6 5	1 0 2 3 4 5 6
↓ 1 0 2 3 4 5 6	1 0 2 3 4 5 6	0 1 2 3 4 5 6

[問 3] 15 点

```
void BubbleSort(void)
{
    int    i,j,flag;

    do
    {
        flag=0;
        for(i=0;i<N-1;i++)
            /* 先頭から順に見ていって */
            {
                if(array[i]>array[i+1])
                    /* 左右の並びがおかしければ */
                    {
                        /* 入れ替える */
                        flag=1;
                        j=array[i];
                        array[i]=array[i+1];
                        array[i+1]=j;
                    }
            }
        /* 1 度も並べ替えをせずに見終わったら終了 */
    } while(flag==1);
}
```

[問 4] 10 点

データ数を N とした場合，バブルソートの計算量は $O(N^2)$ である．これは，データ数の 2 乗に比例して，計算量が増加することを表している．

2 サーチ

〔問 1〕 10 点

バイナリーサーチでは、目的のデータの場所を、予めソートされた規則的に並んでいるデータから探す。候補となっているデータ列の真ん中に位置するデータと比較し、それより前にあるのか後ろにあるのかを捜す。ここで、データの存在する候補が半分になり、この候補の真ん中のデータと比較することにより、候補をさらに半分にすることができる。同じことを繰り返し、候補を $1/2$ ずつにして、探索する方法をバイナリーサーチである。

〔問 2〕 12 点

バイナリーサーチでは、1 回の計算 (比較) で検索する範囲は $1/2$ になる。したがって、 α 回のループでその範囲が 1 になれば計算が終了する。データの個数を N として、式で表すと、

$$N \left(\frac{1}{2} \right)^\alpha = 1$$

となる。これから、計算回数 α は、次のようにして求められる。

$$\left(\frac{1}{2} \right)^\alpha = \frac{1}{N}$$

$$2^\alpha = N$$

$$\log_2 2^\alpha = \log_2 N$$

$$\alpha \log_2 2 = \log_2 N$$

$$\alpha = \log_2 N$$

計算回数 α は $\log_2 N$ となることが示され、計算のオーダーは $O(\log_2 N)$ と表現される。

〔問 3〕 各 2 点

ア d

イ c

ウ b

エ a

3 リスト

[問 1] 5 点

```
typedef struct tagListNode
{
    struct tagListNode *prev; /* 前の要素へのポインタ */
    struct tagListNode *next; /* 次の要素へのポインタ */
    int data; /* この要素がもっているデータ */
} ListNode;
```

4 C 言語プログラムテクニック

[問 1] 5 点

```
array=
(int *)malloc(sizeof(int)*count);
```

[問 2] 5 点

```
array[3]=999;
あるいは,
*(array+3)=999;
```

[問 3] 5 点

```
newnode->data=buf;
```

5 応用問題

[問 1] 5 点

比較の回数は、それぞれ以下ようになる。

- a[0]～[N-1] から最小値を捜すための比較の回数は、 N 回である。
- a[1]～[N-1] から最小値を捜すための比較の回数は、 $N - 1$ 回である。
- a[2]～[N-1] から最小値を捜すための比較の回数は、 $N - 2$ 回である。
- 最後の a[2]～[N-1] から最小値を捜すための比較の回数は、2 回である。

したがって、比較の回数 α は

$$\begin{aligned}\alpha &= 2 + 3 + 4 + \cdots + N \\ &= \sum_{i=1}^N i - 1 \\ &= \frac{N(N+1)}{2} - 1\end{aligned}$$

となる。

プログラムの書き方により、比較の回数は変化する。考え方がただしければ、正解とする。