

情報構造論 前期中間試験 (出題: 齊藤) 番号:

4EI 2005/06/14,(1/2) 氏名:

以下の5つの設問の中から4つを選んで回答せよ。5つの設問すべてに回答してある場合には、採点の高い4題を加算する。

1 再帰呼び出しの理解確認 (× 25)

以下の再帰呼び出しを含むプログラムを『処理 (1)』の様に呼び出した場合の実行順序を下線部の記号を用いて表せ。

```
void foo( int a[] , int size )
{   if ( size > 1 ) {
    ~~~~~(A)
        int i , m , tmp ;
        for( i = 0 , m = size - 1 ; i < size - 1 ; i++ )
            ~~~~~(B) ~~~~~(C) ~~~(D)
            if ( a[ i ] > a[ m ] )
                ~~~~~(E)
                m = i ;
                ~~~~~(F)
            tmp = a[ m ] ;
            ~~~~~(G)
            a[ m ] = a[ size - 1 ] ;
            ~~~~~(H)
            a[ size - 1 ] = tmp ;
            ~~~~~(I)
            foo( a , size - 1 ) ;
            ~~~~~(J)
        }
}

int b[ 3 ] = { 4 , 2 , 8 } ;
int c[ 5 ] = { 10 , 3 , 6 , 7 , 8 } ;
void main()
{
    int i ;
    foo( b , 3 ) ; /* 処理 (1) */
    foo( c , 5 ) ; /* 処理 (2) */
}
```

2 再帰方程式 (× 25)

前のプログラムfoo()の処理速度を、求めたい。ただし、データ件数sizeを N で表す事。

1. 処理速度 $T(N)$ を、再帰方程式にて示せ。速度を表す定数は、必要に応じて用いる事。
2. 再帰方程式を解き、処理速度の一般式を示せ。ただし厳密な証明は不要。
3. 一般式から処理速度をオーダ記法で表現せよ。

3 処理速度の予測 (× 25)

1. 以下の式で示される処理速度を、オーダ記法によって表せ。

$$T(N) = T_A + T_B \times \frac{N(N+1)(2N+1)}{6}$$

2. 処理速度が以下のオーダで示されるプログラムがあったとする。このプログラムがデータ件数 100 において、200[msec]の時間がかかった。この時、データ件数が 10000 件の場合、処理速度を予測せよ。

$$O(\sqrt{N} \log N)$$

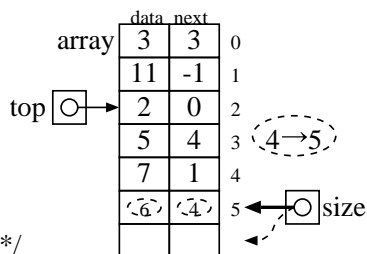
4 疑似リスト構造 (× 25)

データと次のデータの配列の添字によりリスト構造を、下右図のようなデータ構造で扱うものとする。このデータ構造において、新たに1件のデータを追加する `insert()` という関数を作りたい。

ただし回答を簡単にするために、同じデータが複数あったり、先頭にデータが入ることはないものとする。

```

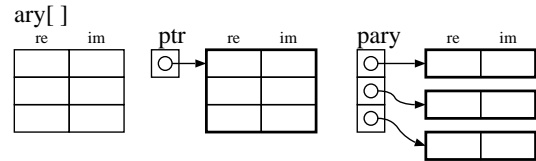
struct List {
    int data ;
    int next ;
} ;
struct List array[ 20 ] = {
    { 3 , 3 } ,
    { 11 , -1 } ,
    { 2 , 0 } ,
    { 5 , 4 } ,
    { 7 , 1 } ,
} ;
int top = 2 ; /* 先頭の場所 */
int size = 5 ; /* データ件数 */
void insert( int x )
{ /* x:挿入したいデータ */
  /* この内部を作成せよ */
}
    
```



例えば `insert(6)` を実行したら図中の丸の内部の様に値が書き変わって欲しい。

5 malloc+free の理解 (× 25)

複素数のデータを構造体の様々な形式にコピーするプログラムを作成した。正しく動作するように下線部を埋めよ。



```
double data[ 3 ][ 2 ] = { /* コピー元のデータ */
    { 1.23 , 4.56 } ,    /* { 実部 , 虚部 } */
    { -2.0 , 4.0 } ,
    { 3.05 , -2.0 } ,
};
struct Complex { /* 複素数を覚える構造体 */
    double re ; /* 実部 */
    double im ; /* 虚部 */
};
void main() {
    struct Complex ary[ 3 ] ;

    ~~~~~ /* ptr の宣言 */ (A)
    struct Complex* pary[ 3 ] ;

    /* 構造体の配列 */
    for( i = 0 ; i < 3 ; i++ ) {
        ~~~~~ = data[ i ][ 0 ] ;
        ary[ i ].im (B) = data[ i ][ 1 ] ;
    }
    /* 動的確保の構造体配列 */

    ptr = ~~~~~ (C)

    if ( ptr != NULL ) {
        for( i = 0 ; i < 3 ; i++ ) {
            ptr[ i ].re = data[ i ][ 0 ] ;
            ~~~~~ = data[ i ][ 1 ] ; (D)
        }
    }
    /* ポインタの配列 */
    for( i = 0 ; i < 3 ; i++ ) {
        pary[ i ] = (struct Complex*)malloc( sizeof( struct Complex ) ) ;

        if ( ~~~~~ ) { (E)
            pary[ i ]->re = data[ i ][ 0 ] ;
            pary[ i ]->im = data[ i ][ 1 ] ;
        }
    }
    /* データを使う処理は省略 */
}
```