

## 平成30年度後期 データ構造とアルゴリズム 期末テスト

※ 各問題中のアルゴリズムを表すプログラムは、変数の宣言が省略されているなど、完全なものではありませんが、適宜、常識的な解釈をしてください。疑問があれば、挙手をして質問してください。

※ 時間計算量をオーダ記法で表せという問題では、入力サイズ  $n$  を無限大に近づけた場合の漸近的な時間計算量を表せということだと考えてください。

**問題1** 入力サイズが  $n$  の問題を解くアルゴリズムの正確な時間計算量が次のような場合、それぞれの漸近的な時間計算量をオーダ記法で表せ。

- |                                    |                              |                                |
|------------------------------------|------------------------------|--------------------------------|
| (1) 3                              | (2) $2n$                     | (3) $4n^2 + 3n$                |
| (4) $5n^3 - n - 2$                 | (5) $2\sqrt{n} + 3n - 1$     | (6) $3\log_{10} n + 2n + 1$    |
| (7) $3\sqrt{n} + 4\log_{10} n - 2$ | (8) $2n\log_{10} n + 5n + 1$ | (9) $3n\log_{10} n + 2n^2 - 4$ |
| (10) $n^{100} + 2^n$               | (11) $2^n + n!$              | (12) $n! + n^{100}$            |

**問題2** 次のプログラムで表される、入力サイズが  $n$  の問題を解くアルゴリズムの時間計算量をオーダ記法で表せ。(2)の  $n$  は2のべき乗の数(整数  $m$  を用いて  $2^m$  と表すことができる数)とする。

(1)  
`x = 0;`  
`for( i = 1; i <= n; i++ ) {`  
`x = x + i;`  
`}`  
`x` の値を出力する;

(2)  
`x = n;`  
`while( x > 1 ) { x = x / 2; }`  
`x` の値を出力する;

(3)  
`for( i = 1; i <= n; i++ ) { x[i] = 0; }`  
`for( i = 1; i <= n; i++ ) {`  
`x[i] = i;`  
`for( j = i+1; j <= n; j++ ) {`  
`x[i] = x[i] + j;`  
`}`  
`}`  
`for( i = 1; i <= n; i++ ) {`  
`x[i]の値を出力する;`  
`}`

**問題3** 次のプログラムはデータを“連結リスト”に格納する。連結リストのレコードは構造体 `rec` で表され、`rec` 中の `d` はデータ、`p` は次のレコードを指すポインタである。連結リストの先頭のレコードを指すポインタ `head` は「何もない」ことを表す値 `NULL` で初期化している。関数 `add` は連結リストの先頭へのデータ `x` の追加、関数 `del` は先頭からのデータの削除、関数 `disp` は連結リスト中の全データの表示を行う。関数 `add` では関数 `malloc` により新たなレコードを生成している。

```
// レコードの構造体
typedef struct rc {
    int d;
    struct rc *p;
} rec;
// 先頭レコードを指すポインタ
rec *head = NULL;

void add( int x )
{
    rec *r;    // 新たなレコードの生成
    r = (rec *)malloc(sizeof(rec));
    r->d = x;    // 追加データの格納
    r->p = head; // (a)
    head = r;   // (b)
}

void del( void )
{
    rec *r;
    if( head != NULL ) {
        r = head->p; // (c)
        free( head ); // 先頭レコードの削除
        head = r;    // (d)
    }
}

void disp( void )
{
    rec *r;
    r = head;
    while( r != NULL ) {
        printf( "%d ", r->d );
        r = r->p;
    }
    printf( "\n" );
}

void main( void )
{
    add( 3 ); add( 7 ); add( 1 );
    disp(); // (★1)
    del();
    disp(); // (★2)
    add( 8 ); add( 4 );
    disp(); // (★3)
    del(); del(); del();
    disp(); // (★4)
}
```

- (1) (a)と(b)の結果として、連結リストに対して新たなレコードがどうされるかを書け。
- (2) (c)と(d)で何が行われているのかを書け。rに何が入るのがわかるように書くこと。
- (3) (★1)から(★4)のそれぞれで表示される全データを書け。表示される順序で書くこと。
- (4) 連結リストの末尾のレコードのポインタ p には何が入るかを書け。

**問題4** 整数  $n$  は  $2$  のべき乗の数とする。このとき、定数  $a$  に対して  $a^n$  を求める次の3つのアルゴリズムを考える。下の問いに答えよ。

```
/* アルゴリズム A */
```

```
powA( n ) {
    x = 1;
    for( i = 0; i < n; i++ ) {
        x = x * a;
    }
    return( x );
}
```

```
/* アルゴリズム B */
```

```
powB( n ) {
    if( n == 1 ) { return( a ); }
    else {
        return( powB( n/2 ) * powB( n/2 ) );
    }
}
```

```
/* アルゴリズム C */
```

```
powC( n ) {
    if( n == 1 ) { return( a ); }
    else {
        p = powC( n/2 );
        return( p * p );
    }
}
```

- (1) アルゴリズム B と C のように関数が内部で自分自身を呼び出すようなアルゴリズムは何と呼ばれるか、名称をかけ。
- (2) それぞれのアルゴリズムの時間計算量をオーダ記法で書け。

**問題5** 次のプログラムは、昇順（小さな値から大きな値の順）の  $n$  個のデータ  $D[0], \dots, D[n-1]$  から“2分探索法”によって値  $x$  を探索する。

```
void binarysearch(  
    double D[],          // データ D[0], ..., D[n-1]  
    int n,              // データの個数  
    double x            // 探索する値  
)  
{  
    left = 0;   right = n - 1;   mid = ( left + right ) / 2;  
    while( left < right ) {  
        if( D[mid] == x ) { break; }  
        else if( D[mid] < x ) { left = ; }  
        else { right = ; }  
        mid = ;  
    }  
    if( D[mid] == x ) { D[mid]を出力; }  
    else { (d) }  
}
```

- (1) 空欄(a)～(c)を埋めよ.
- (2) (d)の箇所が実行されるのはどのような場合か答えよ.
- (3) このアルゴリズムの時間計算量をオーダ記法で表せ.

**問題6** 次のプログラムは、配列  $D[0], \dots, D[n-1]$  中の  $n$  個のデータを“選択ソート”によって昇順に並べ変える。関数 `swap` は2つの引数のデータを交換するものである。下の問いに答えよ。

```
void selectionsort( int n, double D[] )
{
    for( i = n-1; i > 0; i-- ) {
        max = D[0];
        max_index = 0;
        for( j = 1; j <= i; j++ ) {
            if( D[j] >= max ) {
                max = D[j];
                (a)
            }
        }
        swap( &(D[max_index]), &( (b) ) );
    }
}
```

- (1) 空欄を埋めよ。
- (2) データの個数が  $n = 5$  で、ソート前の初期状態として配列  $D[0], \dots, D[4]$  に 9, 1, 5, 3, 7 がこの順序で入っていたとする。変数  $i$  による外側の `for` ループの繰り返しが、変数  $i$  が 4, 3, 2, 1 の値で順に実行されるとき、各値での実行が終了した直後の配列  $D$  中のデータを書け。
- (3) このアルゴリズムの時間計算量をオーダ記法で表せ。

**問題7** 次のプログラムは、配列  $D[0], \dots, D[n-1]$  中の  $n$  個のデータを“挿入ソート”によって昇順に並べ変える。下の問いに答えよ。

```
void insertionsort( int n, double D[] )
{
    for( i = 1; i < n; i++ ) {
        x = D[i];
        j = i;
        while( ( D[j-1] > x ) && ( (a) ) ) {
            (b)
            j = j - 1;
        }
        D[j] = x;
    }
}
```

- (1) 空欄を埋めよ。
- (2) データの個数が  $n = 5$  で、ソート前の初期状態として配列  $D[0], \dots, D[4]$  に 9, 1, 5, 3, 7 がこの順序で入っていたとする。変数  $i$  による外側の `for` ループの繰り返しが、変数  $i$  が 1, 2, 3, 4 の値で順に実行されるとき、各値での実行が終了した直後の配列  $D$  中のデータを書け。
- (3) このアルゴリズムの時間計算量をオーダ記法で表せ。

**問題8** 次のプログラムは，“ヒープ”にデータを追加する関数 `push_heap` とヒープから最大値を取り出す関数 `delete_maximum` である。ヒープの2分木を配列 `T` で表し，変数 `size` はそのサイズである。また，関数 `swap` は2つの引数のデータを交換する。下の問いに答えよ。

```
void push_heap( double T[], /* ヒープを表す配列 T[1], ..., T[size] */
               double x /* 追加するデータ */ )
{
    size++;
    T[size] = x; // データを最後に追加
    k = size;
    while( ( k > 1 ) && (  ) ) {
        swap( &(T[k]), &(T[k/2]) ); k = ;
    }
}

double delete_maximum( double T[] /* ヒープを表す配列 T[1], ..., T[size] */ )
{
    T[1]を出力;
    T[1] = T[size]; T[size]を空にする; // 最後のデータを根に移動
    size--; k = 1;
    while( 2*k <= size ) { // 子を持つかどうかを判定
        if( 2*k == size ) {
            if(  ) {
                swap( &(T[k]), &(T[2*k]) ); k = ;
            }
            else { アルゴリズムを終了; }
        }
        else {
            if(  ) { big = 2*k; }
            else { big = 2*k+1; }
            if( T[k] < T[big] ) {
                swap( &(T[k]), &(T[big]) ); k = ;
            }
            else { アルゴリズムを終了; }
        }
    }
}
```

(1) 空欄を埋めよ。

(2) ヒープのサイズが `size = 10` であり，配列 `T[1], ..., T[10]` の値が  
39, 23, 24, 17, 6, 1, 2, 9, 11, 5

である場合，次のそれぞれについて答えよ。

(2-1) ヒープにデータ `x = 31` を追加するために関数 `push_heap` を実行する。この実行後に `T[1], ..., T[11]` に格納されている値をその順序で答えよ。

(2-2) ヒープから最大値を取り出す関数 `delete_maximum` を実行する。この実行後に `T[1], ..., T[9]` に格納されている値をその順序で答えよ。

**問題9** 次のプログラムは, `quicksort(D, 0, n-1)` を実行することで, 配列 `D[0], ..., D[n-1]` 中の  $n$  個のデータを“クイックソート”によって昇順に並べ変える. なお, プログラム中の変数の宣言は省略している. 下の問いに答えよ.

```
void quicksort(
    double D[],          // データ D[left], ..., D[right]
    int left,           // ソートの対象とする配列 D の左端の位置
    int right          // ソートの対象とする配列 D の右端の位置
)
{
    if( left < right ) {
        pivot_index = partition( D, left, right );          // (a)
        quicksort( D, left, pivot_index-1 );                // (b)
        quicksort( D, pivot_index+1, right );                // (c)
    }
}
```

(1) データの個数が  $n = 10$  で, 配列 `D[0], ..., D[9]` に

`35, 21, 4, 49, 55, 19, 12, 32, 24, 42`

がこの順序で入っている状態で, 関数 `quicksort` が `left = 0, right = 9` として呼び出されたとする. 基準値を `D[0] = 35` とした場合, (a) で実行された関数 `partition` が戻り値 (返り値) として変数 `pivot_index` に渡す値を答えよ.

(2) (1) の後, (b) のために関数 `quicksort` に渡される `D[left], ..., D[pivot_index-1]` に入っているデータをすべて答えよ. 順序は問わない.

(3) (2) の後, (c) のために関数 `quicksort` に渡される `D[pivot_index+1], ..., D[right]` に入っているデータをすべて答えよ. 順序は問わない.

(4) データが  $n$  個のときの上記のプログラムの時間計算量  $T(n)$  は次式であらわされる.

$$T(n) = cn + T(n_l) + T(n_r)$$

右辺の  $c$  は定数であり,  $cn$  は (a),  $T(n_l)$  は (b),  $T(n_r)$  は (c) の時間計算量である.

(4-1)  $n_l$  と  $n_r$  が何をあらわすかを書け. また,  $n, n_l, n_r$  の関係を書け.

(4-2) `D[left]` から `D[right]` の中からどのような値を基準値として選ぶ場合に  $T(n)$  が **最悪時間計算量** になるかを書け. また, このときの  $n_l, n_r$  の値を書け. さらに, その最悪時間計算量を  $n$  に関するオーダ記法で書け.

(4-3) `D[left]` から `D[right]` の中からどのような値を基準値として選ぶ場合に  $T(n)$  が **最良時間計算量** になるかを書け. また, このときの  $n_l, n_r$  の値を書け (近似値でよい). さらに, その最良時間計算量を  $n$  に関するオーダ記法で書け.

**問題10** 次のプログラムは、mergesort(D, 0, n-1)を実行することで、配列 D[0], ..., D[n-1]中の n 個のデータを“マージソート”によって昇順に並べ変える。なお、プログラム中の変数の宣言は省略している。下の問いに答えよ。

```
void mergesort(
    double D[],          // データ D[left], ..., D[right]
    int left,           // ソートの対象とする配列 D の左端の位置
    int right          // ソートの対象とする配列 D の右端の位置
)
{
    mid = ( left + right ) / 2;                // (★1)
    if( left < mid ) { mergesort( D, left, mid ); } // (★2)
    if( mid+1 < right ) { mergesort( D, mid+1, right ); } // (★3)
    merge( D, left, mid, right );
}

void merge(
    double D[],        // データ D[left], ..., D[right]
    int left,         // ソートの対象とする配列 D の左端の位置
    int mid,          // ソートの対象とする配列 D の中央の位置
    int right        // ソートの対象とする配列 D の右端の位置
)
{
    x = left;    y = mid + 1;
    for( i = 0; i <= right-left; i++ ) {
        if( x == mid+1 ) {  (a)  ;  (b)  ; }
        else if( y == right+1 ) {  (c)  ;  (d)  ; }
        else if( D[x] <= D[y] ) {  (e)  ;  (f)  ; }
        else {  (g)  ;  (h)  ; }
    }
    for( i = 0; i <= right - left; i++ ) { D[left+i] = M[i]; }
}
```

(1) 空欄(a)～(h)を埋めてプログラムを完成させよ。

(2) データの個数が n = 10 で、配列 D[0], ..., D[9]に

35, 21, 4, 49, 55, 19, 12, 32, 24, 42

がこの順序で入っている状態で、関数 mergesort が left = 0, right = 9 で呼び出されたとする。

(2-1) (★1)で mid に代入される値を答えよ。

(2-2) (★2)の関数 mergesort の実行後の D[left], ..., D[mid]中の値をその順序で答えよ。

(2-3) (★3)の関数 mergesort の実行後の D[mid+1], ..., D[right]中の値をその順序で答えよ。

平成30年度後期 データ構造とアルゴリズム 期末テスト 解答用紙

平成 \_\_\_\_\_ 年度入学 学籍番号 \_\_\_\_\_ 氏名 \_\_\_\_\_

**問題1**

- (1) \_\_\_\_\_ (2) \_\_\_\_\_ (3) \_\_\_\_\_
- (4) \_\_\_\_\_ (5) \_\_\_\_\_ (6) \_\_\_\_\_
- (7) \_\_\_\_\_ (8) \_\_\_\_\_ (9) \_\_\_\_\_
- (10) \_\_\_\_\_ (11) \_\_\_\_\_ (12) \_\_\_\_\_

**問題2**

- (1) \_\_\_\_\_
- (2) \_\_\_\_\_
- (3) \_\_\_\_\_

**問題3 (1)**

(2)

(3)

(★1) \_\_\_\_\_ (★2) \_\_\_\_\_

(★3) \_\_\_\_\_ (★4) \_\_\_\_\_

(4)

**問題4 (1)**

(2)

A \_\_\_\_\_ B \_\_\_\_\_

C \_\_\_\_\_

**問題5 (1)**

(a) \_\_\_\_\_ (b) \_\_\_\_\_

(c) \_\_\_\_\_

(2)

(3)

**問題6 (1)**

(a) \_\_\_\_\_ (b) \_\_\_\_\_

(2)

i	D[0]	D[1]	D[2]	D[3]	D[4]
初期状態	9	1	5	3	7
4					
3					
2					
1					

(3)

**問題7 (1)**

(a) \_\_\_\_\_ (b) \_\_\_\_\_

(2)

i	D[0]	D[1]	D[2]	D[3]	D[4]
初期状態	9	1	5	3	7
1					
2					
3					
4					

(3)

**問題8** (1)

(a) \_\_\_\_\_ (b) \_\_\_\_\_

(c) \_\_\_\_\_ (d) \_\_\_\_\_

(e) \_\_\_\_\_ (f) \_\_\_\_\_

(2) (2-1)

\_\_\_\_\_

(2-2)

\_\_\_\_\_

**問題9** (1)

\_\_\_\_\_

(2)

\_\_\_\_\_

(3)

\_\_\_\_\_

(4) (4-1)

**問題9** (4-2)

\_\_\_\_\_

(4-3)

**問題10** (1)

(a) \_\_\_\_\_ (b) \_\_\_\_\_

(c) \_\_\_\_\_ (d) \_\_\_\_\_

(e) \_\_\_\_\_ (f) \_\_\_\_\_

(g) \_\_\_\_\_ (h) \_\_\_\_\_

(2) (2-1)

\_\_\_\_\_

(2-2)

\_\_\_\_\_

(2-3)

\_\_\_\_\_

