

平成28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 サイズ n の配列 $s[0], \dots, s[n-1]$ を用いて、次の2つの関数でスタックを実現する。下の問いに答えよ。

```
/* スタック s に対して、データ x を格納する */
push( S, x ) {
    top = ;
    if(  ) {
        "オーバーフロー"と出力;
    }
    else {
        S[top] = x;
    }
}
```

```
/* スタック s からデータを取り出し、そのデータを出力する */
pop( S ) {
    if(  ) {
        "アンダーフロー"と出力;
    }
    else {
        S[top]の値を出力;
        top = ;
    }
}
```

(1) 関数中の空欄を埋めよ。

(2) スタックを表す配列 s を空に初期化する場合、変数 top をどのような値にすればよいか答えよ。

(3) 空のスタック s に対して以下の操作を順番に実行した。

push($S, 4$) \rightarrow push($S, 3$) \rightarrow push($S, 8$) \rightarrow pop(S)
 \rightarrow pop(S) \rightarrow push($S, 7$) \rightarrow push($S, 1$) \rightarrow pop(S)

(3-1) 1, 2, 3回目の pop で出力される値をそれぞれ答えよ。

(3-2) 配列のサイズが $n = 4$ の場合、全ての操作の終了後に配列の各要素に入っている値を答えよ。

ただし、pop で出力した値は配列に残らないものとし、値が入っていない場合には \times と答えよ。

問題4 サイズ n の配列 $Q[0], \dots, Q[n-1]$ を用いて、次の2つの関数でキューを実現する。下の問いに答えよ。

```
/* キューQに対して、データ x を格納する */
enqueue( Q, x ) {
    Q[right] = x;
    right = ;
    if( right == n ) {  }
    if( left == right ) {
        "オーバーフロー"と出力;
    }
}
```

```
/* キューQ からデータを取り出し、そのデータを
出力する */
dequeue( Q ) {
    if( left == right ) {
        "アンダーフロー"と出力;
    }
    else {
        Q[left]の値を出力;
        left = ;
        if( left == n ) {  }
    }
}
```

- (1) 関数中の空欄を埋めよ。
- (2) キューを表す配列 Q を空に初期化する場合、変数 $left$ と $right$ をどのような値にすればよいか答えよ。
- (3) 空のキュー Q に対して以下の操作を順番に実行した。
 - $enqueue(Q, 4) \rightarrow enqueue(Q, 3) \rightarrow enqueue(Q, 8) \rightarrow dequeue(Q)$
 - $\rightarrow dequeue(Q) \rightarrow enqueue(Q, 7) \rightarrow enqueue(Q, 1) \rightarrow dequeue(Q)$
- (3-1) 1, 2, 3回目の $dequeue$ で出力される値をそれぞれ答えよ。
- (3-2) 配列のサイズが $n = 4$ の場合、全ての操作の終了後に配列の各要素に入っている値を答えよ。ただし、 $dequeue$ で出力した値は配列に残らないものとし、値が入っていない場合には \times と答えよ。

問題5 完全2分木について、次の問いに答えよ。なお、完全2分木では、木の高さが h のとき、根のレベルは 0 であり、葉のレベルは $h-1$ である。下の問いに答えよ。

- (1) 木の高さ h を用いて、葉の個数を表せ。オーダ記法ではなく、正確な値を書くこと。
- (2) 木の高さ h を用いて、全節点（根と葉も含む）の個数を表せ。オーダ記法ではなく、正確な値を書くこと。

問題6 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“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8 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9 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“ヒープソート”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。

```
// ヒープソート
void heapsort( int n, /* データの個数 */
              double D[] /* データ D[0], ..., D[n-1] */ )
{
    size = 0; // ヒープを表す2分木の配列のサイズを初期化
    for( i = 0; i < n; i++ ) { push_heap( T, D[i] ); }
    for( i = n-1; i >= 0; i-- ) { D[i] = delete_maximum( T ); }
}

// ヒープにデータを追加する
void push_heap( double T[], /* ヒープを表す2分木の配列 T[1], ..., T[N] */
               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 ) && ( T[k] > T[k/2] ) ) { // (1)
        swap( &(T[k]), &(T[k/2]) ); k = k/2; // (1)
    }
}

// ヒープから最大値を取り出す
double delete_maximum( double T[] /* ヒープを表す2分木の配列 T[1], ..., T[N] */ )
{
    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 ) { // (2-1)
            if( T[k] < T[2*k] ) {
                swap( &(T[k]), &(T[2*k]) ); k = 2*k;
            }
            else { アルゴリズムを終了; }
        }
        else { // (2-2)
            if( T[2*k] > T[2*k+1] ) { big = 2*k; } // (3)
            else { big = 2*k+1; } // (3)
            if( T[k] < T[big] ) { // (4)
                swap( &(T[k]), &(T[big]) ); k = big; // (4)
            }
            else { アルゴリズムを終了; }
        }
    }
}
}
```

- (1) プログラム中の(1)について、 $T[k]$ と $T[k/2]$ の関係、および、この処理の目的を説明せよ。
- (2) プログラム中の(2-1)と(2-2)はどんな条件による分岐なのか説明せよ。
- (3) プログラム中の(3)について、 $T[2*k]$ と $T[2*k+1]$ の関係、および、この処理の目的を説明せよ。
- (4) プログラム中の(4)について、 $T[k]$ と $T[big]$ の関係、および、この処理の目的を説明せよ。
- (5) 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10 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。

問題11 次の表に示す4種類の牛肉があり、これらは分割してもよい。これらを重さの限界が 40kg の袋に入れるとき、価格の総和を最大にするためには、それぞれの牛肉を何 kg ずつ入れればよいか、グリーディ法により計算せよ。

牛肉	重さ (kg)	価格 (万円)
A	30	20
B	10	10
C	30	40
D	20	50

平成28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)

(a) _____ (c) _____

(b) _____ (d) _____

(2)

(3-1) 1 : _____ 2 : _____ 3 : _____

(3-2)

S[0]	S[1]	S[2]	S[3]

問題 4 (1)

(a) _____ (c) _____

(b) _____ (d) _____

(2)

(3-1) 1 : _____ 2 : _____ 3 : _____

(3-2)

Q[0]	Q[1]	Q[2]	Q[3]

問題 5 (1)

(2)

問題 6 (1)

(2)

A _____ B _____

C _____

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
4					
3					
2					
1					

(3)

問題 8 (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9 (1)

(2)

(3)

(4)

(5)

問題10 (1)

(2)

(3)

問題11