令和２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　サイズ n の配列 S[0]，…，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 = [ (a) ];
    if( [ (b) ] ) {
        “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( [ (c) ] ) {
        “アンダーフロー”と出力;
    }
    else {
        S[top]の値を出力;
        top = [ (d) ];
    }
}
```

（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) → push(S, 3) → push(S, 8) → pop(S)

　　→ pop(S) → push(S, 7) → push(S, 1) → 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 × と答えよ．

問題３　サイズ n の配列 Q[0], …, 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 = [   (a)   ];
    if( right == n ) { [   (b)   ] }
    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 = [   (c)   ];
        if( left == n ) { [   (d)   ] }
    }
}
```

（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) → enqueue(Q, 3) → enqueue(Q, 8) → dequeue(Q)
→ dequeue(Q) → enqueue(Q, 7) → enqueue(Q, 1) → 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 × と答えよ．

問題４　**“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．

（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], ..., 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 = [  (a)  ]; }
        else { right = [  (b)  ]; }
        mid = [        (c)        ];
    }
    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最良時間計算量，最悪時間計算量，平均時間計算量をオーダ記法で表せ．

問題６ 次の関数 selectionsort は **“選択ソート”**，関数 insertionsort は **“挿入ソート”** によって，配列 D[0]，...，D[n-1]中の n 個のデータを昇順に並べ変える．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[] )    // 選択ソート
{
    for( i = n-1; i > 0; i-- ) {
        max = D[0];
        max_index = 0;
        for( j = 1; j <= i; j++ ) {
            if( D[j] >= max ) {    // ← ①
                max = D[j];
               [     (a)     ]
            }
        }
        swap( &(D[max_index]), &( [ (b) ] ) );
    }
}

void insertionsort( int n, double D[] )    // 挿入ソート
{
    for( i = 1; i < n; i++ ) {
        x = D[i];
       [  (c)  ]
        while( ( j > 0 ) && ( D[j-1] > x ) ) {    // ← ②
           [     (d)     ]
            j = j - 1;
        }
        D[j] = x;
    }
}
```

（１）それぞれの関数の空欄を埋めよ．

（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]，…，D[4]に下の(A)と(B)の値がその順序で入っていたとする．(A)と(B)のそれぞれの場合について，それぞれの関数が変数 i による外側の for ループの繰り返しで i の値を変えながら実行されるとき，変数 i の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．また，変数 i の各値において，選択ソートでは①中の比較 D[j] >= max，挿入ソートでは②中の比較 D[j-1] > x が実行される回数を書け．

(A) 9, 1, 5, 3, 7　　　　　(B) 9, 7, 5, 3, 1

（３）それぞれの関数について，最良時間計算量，最悪時間計算量，平均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], ..., 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８ 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９ 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;                                    // (a)
    if( left < mid ) { mergesort( D, left, mid ); }                // (b)
    if( mid+1 < right ) { mergesort( D, mid+1, right ); }          // (c)
    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 ) { M[i] = D[y]; y++; }             // (d)
        else if( y == right+1 ) { M[i] = D[x]; x++; }      // (e)
        else if( D[x] <= D[y] ) { M[i] = D[x]; x++; }      // (f)
        else { M[i] = D[y]; y++; }                         // (g)
    }
    for( i = 0; i <= right-left; i++ ) { D[left+i] = M[i]; }
}
```

（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 mergesort が left = 0, right = 9 で呼び出されたとする．

（1－1）(a)で mid に代入される値を答えよ．

（1－2）(b)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．

（1－3）(c)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．

（2）関数 merge に渡される D[left], …, D[mid]を左データ列，D[mid+1], …, D[right]を右データ列と呼ぶこととする．(d)から(g)のそれぞれが実行されるのはどのような場合か，左データ列と右データ列という用語を用いて答えよ．

令和２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・令和　　　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）

(a)　　　　(c)

(b)　　　　(d)

（２）

（３－１）１：　　２：　　３：

（３－２）

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

問題３（１）

(a)　　　　(c)

(b)　　　　(d)

（２）

（３－１）１：　　２：　　３：

（３－２）

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

問題４

（１）　　　　（２）

問題５（１）

(a)　　　　(b)

(c)

（２）

（３）最良：　　　　最悪：　　　　平均：

問題６（１）

(a)　　　　(b)

(c)　　　　(d)

（２）選択ソート　(A)

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

選択ソート　(B)

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

挿入ソート　(A)

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

挿入ソート　(B)

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

問題６（３）

	最良	最悪	平均
選択ソート			
挿入ソート			

問題７（１）

（２）

（３）

（４）

（５）

問題８（１）

（２）

（３）

（４）（４－１）

問題８（４－２）

（４－３）

問題９（１－１）

（１－２）

（１－３）

（２）(d)

(e)

(f)

(g)