

科目名	区分	学籍番号	氏名	得点
データ構造とアルゴリズム	中間試験			/130

※読んで分かるように回答すること。問題との対応が付くように答案用紙に記入すること。160 点満点であり、すべての問題に解答する必要はない。100 点以上になるように問題を選択して解いてよい。多く正解すれば成績評価に加味する。なお、□ 内の数値は配点を示す。

問題 1 次の式のオーダー記法を示せ。例えば、 $10n^2 + 100n$ は $O(n^2)$ と示せばよい。[10]

- (1) $5 + 0.001n^3 + 0.025n$ (2) $500n + 100n^{1.5} + 50n \log_{10} n$ (3) $n^2 \log_2 n + n(\log_2 n)^2$

問題 2 入力を配列 $A[] = \{1, 5, 7, 3, 4, 8, 2, 6\}$ とし、マージソートの動作を図示せよ。解答は図だけでよい。[10]

問題 3 配列 $A[p..r]$ に対するマージソートの擬似コードを次に示す。補助手続き $\text{MERGE}(A, p, q, r)$ について、次の設問に答えよ。[30]

- (1) 補助手続き $\text{MERGE}(A, p, q, r)$ にある空欄 A～F に入るコードを示せ。
 (2) 番兵の役割について解説せよ。
 (3) 第 1～10 行は初期設定である。ループ不変条件を次に示す。空欄アとイを示せ。

第 11～17 行の **for** 文の各繰返しの開始時点では、部分列 $A[\text{ア}]$ には、 $L[1..n_1 + 1]$ と $R[1..n_2 + 1]$ が含む要素全体の中で小さい方から **イ** 個の要素がソートされた順序で置かれている。また、 $L[i]$ と $R[j]$ は配列 L と R のまだ A に書き戻されていない要素の中で最小の要素である。

- (4) 終了条件を次に示す。空欄ウ～カを示せ。

$k = \text{ウ}$ が成立したときにループは停止する。ループ不変式から、部分列 $A[\text{エ}]$ には、 $L[1..n_1 + 1]$ と $R[1..n_2 + 1]$ に含まれる全要素の中で小さい方から **オ** 個の要素がソートされた順序で置かれている。一方、配列 L と R は全部で **カ** 個の要素を含む。したがって、2 個の最大要素を除くすべての要素が A に既に書き戻されていて、残された 2 つの最大要素は番兵である。

アルゴリズム $\text{MERGESORT}(A, p, r)$

```

1 if  $p < r$ 
2    $q = \lfloor (p + r)/2 \rfloor$ 
3    $\text{MERGESORT}(A, p, q)$ 
4    $\text{MERGESORT}(A, q + 1, r)$ 
5    $\text{MERGE}(A, p, q, r)$ 
```

補助手続き $\text{MERGE}(A, p, q, r)$

```

1  $n_1 = \text{A}$ 
2  $n_2 = r - q$ 
3 for  $i = 1$  to  $n_1$ 
4    $L[i] = A[\text{B}]$ 
5 for  $j = 1$  to  $n_2$ 
6    $R[j] = A[q + j]$ 
7  $L[n_1 + 1] = \infty$ 
8  $R[n_2 + 1] = \infty$ 
9  $i = 1$ 
10  $j = 1$ 
11 for  $k = p$  to  $r$ 
12   if  $L[i] \leq R[j]$ 
13      $\text{C}$ 
14      $\text{D}$ 
15   else
16      $\text{E}$ 
17      $\text{F}$ 
```

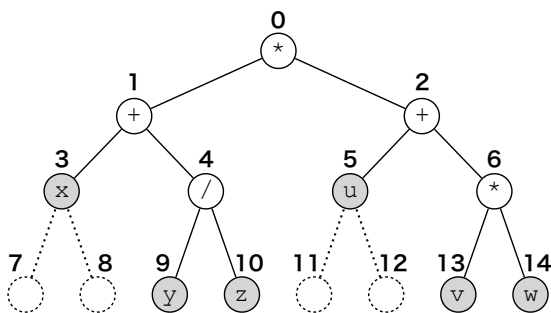
問題 4 下図は、ポインタ p が指している値 15 のレコード（要素）を削除する前後の様子を示している。リストのデータ構造は `struct LIST { int data; struct LIST *next; }` である。削除する C プログラムを示せ。ただし、ポインタ p が最後尾のレコードを指している場合でも正しく動作すること。[30]



問題 5 動的ハッシュ法では、データの追加と削除ができる。動的ハッシュ法に対応したデータ構造を次に示す。動的ハッシュ法でのデータの削除と探索の C プログラムを示せ。関数名は、`int Delete(int data)` と `int Search(int data)` とせよ。[30]

```
1 typedef enum { EMPTY, USE, DELETE } element_stat;
2 typedef struct {
3     element_stat stat;
4     int data;
5 } HashTable;
6 HashTable htable[100] ;
```

問題 6 ヒープの応用として、式 $((x + (y/z)) * (u + (v * w)))$ の 2 分木に変換するプログラムを考える。入力は、`char exp[] = "+x/u*__yz__vw"`; として、下図のような 2 分木を生成するプログラムを示せ。下図に示した 2 分木のデータ構造を使うこと。関数名は `MakeTree(char e[])` とせよ。配点 30



```
1 #define MAXNODE 20
2 char exp[MAXNODE] = "+x/u*__yz__vw";
3 typedef struct BSTnode {
4     char op;
5     struct BSTnode *lson, *rson;
6 } BSTnode;
7 BSTnode node[MAXNODE];
```

問題 7 問題 6 の 2 分木へのポインタ `node` を引数とし、中置記法を出力する C プログラムを示せ。ただし、カッコは出力する必要はない。[10]

問題 8 基本的なデータ構造であるスタックとキューの次のプログラムを示せ。必要な変数は適宜定義すれば良い。[10]

```
1 void InitStack() ;
2 void Push(int x) ;
3 int Pop() ;
4 int EmptyStack() ;
5
6 void InitQueue() ;
7 void Enqueue(int x) ;
8 int Dequeue() ;
9 int EmptyQueue() ;
```

