

## コンピュータサイエンスとプログラミング (講義)

## 試験問題

解答用紙に関する注意: 問 1, 2 は青色の解答用紙, 問 3, 4 は茶色の解答用紙に記入すること. なお, テキスト, ノート, その他図書, 資料の持ち込み不可.

1 以下の問いに答えよ. (25 点)

(1) 次のオーダ記法を簡略化せよ. 但し, 解答は最も簡略化した式で記述すること.

(a)  $O(2n+3n^3+5)$

(b)  $O(10 \log n) + O(n \log n - 15n)$

(c)  $O(\log n)O(n^2+100)$

(d)  $O\left(\left(\frac{3}{2}\right)^{n-1} - \log n\right)$

(2) 以下の項目について簡潔に説明せよ. 必要であれば図を用いて説明しても良い.

(a) アルゴリズムと手続き

(b) リスト, 待ち行列, 優先度付き待ち行列

(c) 問題と問題例

(3) 以下のプログラムは, ポインタによるスタック操作 (PUSH, POP) の関数である. 関数 push は要素  $x$  をスタックの先頭に挿入するための関数であり, 関数 pop はスタックの先頭要素を削除するための関数である. ただし, 両関数の引数 init は, スタックの先頭を指すポインタとする. 以下の空欄 (I), (II) を埋めよ.

```
struct cell {int element; struct cell *next;};

struct cell *push(int x, struct cell *init)
{
    struct cell *q, *r;
    r=(struct cell *)malloc(sizeof(struct cell));
    q=init;
    init=r;
    r->element= x;
    r->next= (I);
    return(init);
}

struct cell *pop(struct cell *init)
{
    struct cell *q;
    if (init!=NULL)
    {
        q=init;
        init= (II);
        free(q);
        return(init);
    }
    else {printf("Error: Stack is empty.\n"); exit(1);}
}
```

2 以下の問いに答えよ。(25 点)

(1) 図 1 の木について、以下の問いに答えよ。

- 図 1 に示す木の高さを答えよ。
- 図 1 に示す木を、配列を用いて実現することを考える。各節点に対応する番号の配列要素に、その節点の親の情報を格納する場合、配列はどのようなになるか図示せよ。ただし、根の位置の配列要素には「-1」を挿入すること。
- 図 1 に示す木を、ポインタを用いて実現することを考える。各節点に対応する番号の配列要素に、その節点の子の情報を連結リストとして格納する場合、配列および連結リストがどのようなになるか図示せよ。ただし、各連結リスト内におけるデータの並び順は特に指定しない。
- 図 1 に示す木を、2 分木を用いて表現することを考える。各節点において、長男を左の子とし、次弟を右の子としたとき、図 1 に示す木を 2 分木として表現せよ。

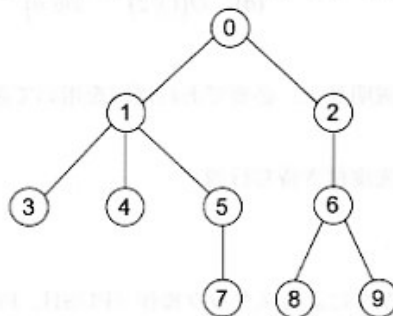


図 1

(2) ハッシュに関して、以下の問いに答えよ。

- 外部ハッシュと内部ハッシュについて、それぞれ簡潔に説明せよ。
- バケット数 5 の外部ハッシュを考える。ハッシュ関数  $h(x) = x \bmod 5$  を用いて、空のハッシュ表に数値データ  $x$  を {23, 48, 35, 4, 10, 58, 42} の順番で格納する場合、ポインタによる連結リストを用いて格納後のハッシュ表を図示せよ。ただし、各連結リストにおいては先頭からデータを追加するものとする。
- 大きさ 7 の配列  $A[i]$  ( $i=0,1,2, \dots, 6$ ) に要素を格納する内部ハッシュを考える。各要素のハッシュ関数  $h(\cdot)$  は表 1 で与えられるものとし、要素  $x$  の  $j$  回目の衝突を回避する新しいハッシュ関数  $h_j(x)$  は、 $h_j(x) = h(x) + j \pmod{7}$  であるとする。ここで、配列  $A$  の全要素は空であるものとする。内部ハッシュ法を適用して 7 つの要素、a,b,c,d,e,f,g をこの順番で挿入した後の配列  $A[i]$  ( $i=0,1,2, \dots, 6$ ) に貯えられている要素を述べよ。

表 1

要素 $x$	a	b	c	d	e	f	g
$h(x)$	1	0	4	0	1	2	3

3

ヒープおよび2分探索木に関する以下の設問に答えよ。(25点)

- (1) 図2に示すヒープに対して、要素2を挿入 (INSERT) するとどのような状態になるかを図示せよ。(図2と同様に、要素の値を含む丸印と実線を用い、最終状態のみを示すこと。)

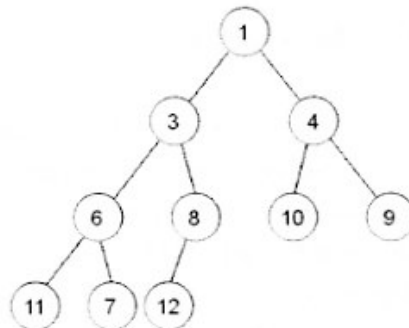


図2

- (2) 図2のヒープ (要素2の挿入前) に対して、最小要素の削除 (DELETETEMIN) を実行するとどのような状態になるかを図示せよ。(最終状態のみを示すこと)
- (3) ヒープに対して DELETETEMIN および INSERT のそれぞれの操作を1回実行するのに要する最悪計算手間 (時間量) をオーダー表記で示せ。ただし、ヒープ内の要素数を  $n$  とする。
- (4) 図3に示す2分探索木に対して、要素19を挿入 (INSERT) するとどのような状態になるかを図示せよ。(図3と同様に、要素の値を含む丸印と実線を用い、最終状態のみを示すこと。)

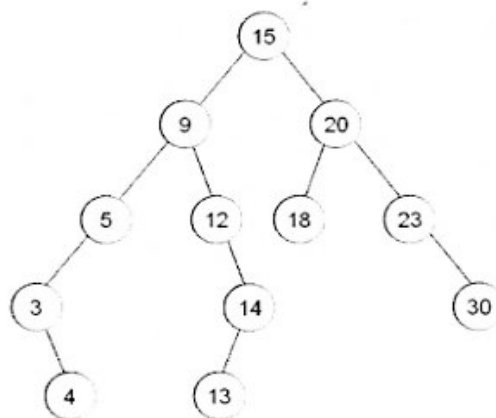


図3

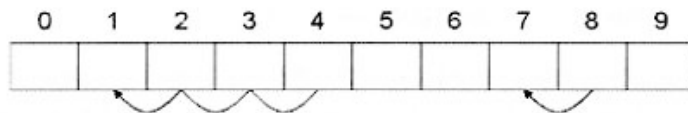
- (5) 図3の2分探索木 (要素19の挿入前) に対して、要素9を削除 (DELETE) するとどのような状態になるかを図示せよ。(最終状態のみを示すこと)
- (6) 2分探索木に対して INSERT, DELETE, MEMBER, MIN のそれぞれの操作を1回実行するのに要する平均計算手間 (時間量) をオーダー表記で示せ。ただし、2分探索木内の要素数を  $n$  とする。

4 ソート（整列）に関する以下の設問に答えよ。（25 点）

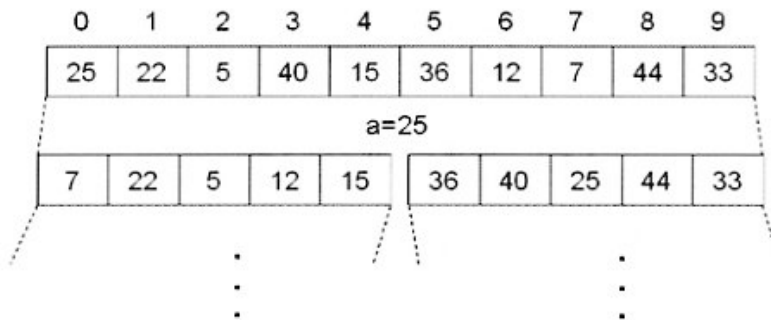
配列 A にデータ 25, 22, 5, 40, 15, 36, 12, 7, 44, 33 が下記のように格納されているものとする。

0	1	2	3	4	5	6	7	8	9
25	22	5	40	15	36	12	7	44	33

- (1) 配列 A をバブルソートにより、昇順にソートするときの要素の変化を、配列の形式で示せ。ただし、バブルソートでは配列の走査が何回か（通常は、（要素数-1）回）行われるため、走査 1 回毎の変化が分かるように、走査が行われる回数分を図示せよ。
- ただし、解答では、配列における要素の値を示すだけでなく、各走査により交換された要素が明確になるよう、以下の例における記法にならって図示すること。なお、この例では、配列番号 8 の要素が配列番号 7 の要素と交換され、配列番号 4 の要素が配列番号 3, 2, 1 の要素と連続的に交換された場合を示している。（この例では要素の値は省略しているが、回答では省略しないこと。）



- (2) 配列 A をクイックソートするときの要素の変化を、配列の形式で示せ。ただし、各グループの軸要素 a は、先頭から見て最初に得られた二つの異なる値のうち大きい方をとるものとする。
- なお、下記は 1 回目のグループ分割の様子を表している。解答では、この記法にならって、ソートが終了するまで、軸要素とグループ分割の様子を図示すること。



- (3) ~~バブルソート~~ バブルソート、バケットソート、ヒープソート、クイックソートのそれぞれの最悪時間量をオーダー表記で示せ。また、クイックソートに関しては、平均時間量も示せ。ただし、整列対象となる順序付き集合の要素数を  $n$  とする。