

(1)~(4)の選択肢:

- (a) p (b) Q (c) Q->front (d) Q->rear (e) p->element (f) p->next (g) p->previous
(h) Q->front->element (i) Q->front->next (j) Q->front->previous (k) Q->rear->element
(l) Q->rear->next (m) Q->rear->previous (n) p->next->element (o) p->next->next
(p) p->next->previous (q) p->previous->element (r) p->previous->next
(s) p->previous->previous (t) Q->front->next->element (u) Q->front->next->next
(v) Q->front->next->previous (w) Q->rear->previous->element (x) Q->rear->previous->next
(y) Q->rear->previous->previous

【問4】2分探索木に関して、以下の問いに答えよ。(12点)

- (1) 空の2分探索木に対して、8個の整数6, 8, 3, 7, 4, 2, 1, 5を左から順に挿入 (INSERT) したときの2分探索木の変化の様子を順を追って示せ。
- (2) (1)で得られた2分探索木から、整数3, 6をこの順番で削除 (DELETE) したときの2分探索木の変化の様子を順を追って示せ。

【問5】 $B=8$ 、ハッシュ関数 $h(x) = x \% B$ (x を B で割った余り)としたとき、8個の整数データ11, 5, 13, 9, 3, 12, 6, 21を左から順に外部ハッシュ法ですべて挿入した後のハッシュ表を求めよ。ただし、挿入には以下に示す図5のプログラム (アルゴリズム自体は授業で説明した外部ハッシュ法そのものである)を用いるものとし、挿入前のハッシュ表 $A[]$ は空 (すべての $A[i] == \text{NULL}$) であるとする。(8点)

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

void insert(int x, struct cell *A[])
{
    struct cell *p, *q, *r;
    int i;
    p = (struct cell *)malloc(sizeof(struct cell));
    p->element = x; p->next = NULL;
    i = h(x); q = A[i];
    if(q == NULL) A[i] = p;
    else {
        while(q != NULL) {
            if(q->element == x) {
                printf("整数 %d はすでに存在するので、挿入しません.\n", x); return;
            } else {
                r = q; q = q->next;
            }
        }
        r->next = p;
    }
}
```

図5: 構造体 struct cell の定義と外部ハッシュ法によるデータの挿入関数 insert

【問6】8個の整数データ5, 2, 1, 8, 7, 4, 3, 6を、次ページの図6~図8に示すクイックソートのアルゴリズム (プリントの図4.E~図4.Gと同じアルゴリズム)を用いて整列することを考える。整列前の配列 A に上記のデータが順番に格納されている (つまり $A[0]=5, A[1]=2, \dots, A[7]=6$ である) とき、関数 quicksort のすべての呼び出し時の配列 A の状態を呼び出しの順番どおりに以下のように記せ。(13点)

quicksort([5,2,1,8,7,4,3,6],0,7) <--- 最初の呼び出し
:
quicksort([1,2,3,4,5,6,7,8],7,7) <--- 最後の呼び出し