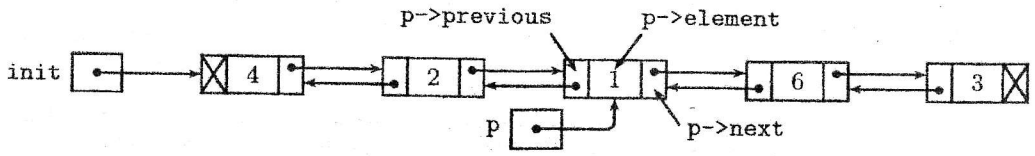
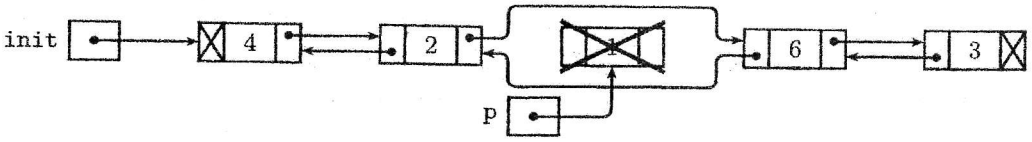
【問3】双方向リスト（直前と直後の両方のセルを指している線形リスト）の中のあるセルへのポインタpが与えられたとき，以下の図に示すように，pが指すセル自身を削除する定数時間のアルゴリズムが実現できる．



(a) pの指すセルを消去する



(b) 消去した後のリストの状態

上記の手法によりリストの要素を削除するプログラム（および構造体 struct cell の定義）を以下に示す．このプログラムの空欄 (1) ～ (4) に入るべき適切な記述を，選択肢 (a) ～ (t) の中から選べ．なお，リストが空（つまり init == NULL）および消去すべきセルへのポインタが空（つまり p == NULL）の場合は考えなくてもよいものとする（20点）．

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

struct cell *delete(struct cell *p, struct cell *init)
{
  if(p == init) init = [ (1) ];  /* リストの先頭を消去するとき */
  else [ (2) ] = [ (1) ];  /* リストの先頭以外を消去するとき */
  if(p->next != NULL)          /* リストの末尾以外を消去するとき */
    [ (3) ] = [ (4) ];
  free(p);
  return init;
}
```

(1)～(4)の選択肢：

(a) p	(b) init	(c) p->element	(d) p->next	(e) p->previous	(f) init->element
(g) init->next	(h) init->previous	(i) p->next->element	(j) p->next->next		
(k) p->next->previous	(l) p->previous->element	(m) p->previous->next			
(n) p->previous->previous	(o) init->next->element	(p) init->next->next			
(q) init->next->previous	(r) init->previous->element	(s) init->previous->next			
(t) init->previous->previous					

【問4】2分探索木に関して，以下の問いに答えよ．(15点)

(1) 図4に示す2分探索木を構成するには，7個の整数データ 1, 3, 5, 7, 9, 11, 13 をどのような順番で挿入（INSERT）すればよいかを答えよ．

(2) 図4の2分探索木に対して，以下の操作を順に適用したときの木の変化の様子を順を追って示せ．

10を挿入 → 7を削除

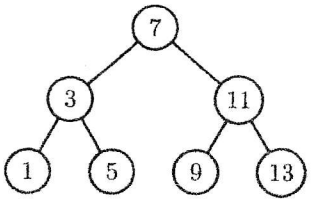


図 4: 完全2分探索木