

集合 A および B の共通集合 C を求める手続き  
(集合は, 要素が昇順に並べられたリストである)

```
typedef struct {
    elementtype element ;
    CELL↑ next ;
} CELL ;

typedef LIST SET ;
すなわち,      typedef struct {
                                CELL↑ head ;
                                CELL↑ last ;
                                } SET ;
```

elementtype の二つの要素  $x, y$  が同じであるか否かを調べる関数  $Same(x, y)$  および  $x$  が  $y$  より大きいかな否かを調べる関数  $GreaterThan(p, q)$  が与えられているものとする.

```
boolean Same(x, y)
    入力 : x, y : elementtype
    出力: Same : boolean
{
    if ( x = y )      return true ;
    else              return false ;
} /* Same */
```

```
boolean GreaterThan(p, q)
    入力 : x, y : elementtype
    出力: GreaterThan : boolean
{
    if ( x > y )      return true ;
    else              return false ;
} /* GreaterThan */
```

```

void Intersection( A, B, C )
    入力 : A: SET(値呼び)
           B: SET(値呼び)
    出力:  C: SET(名前呼び)
    /* 以下で用いている関数は, 全てリストに対するものである */
    /* Makenull や Insert などは集合と同じ名前になってしまったので, 混同しないように */
{
0-1°   p := First( A ) ;           /* p はリスト A の先頭の位置 */
0-2°   q := First( B ) ;           /* q はリスト B の先頭の位置 */
0-3°   r := Makenull( C ) ;        /* r はリスト C の先頭の位置 */
1°     while ( p Tail( A ) and q Tail( A ) ) {
1-1°       if ( Same( Retrieve( p, A ), Retrieve( q, B ) ) ) {
1-1-1°         Insert( Retrieve( p, A ), Tail( A ), C ) ;
1-1-2°         p := Next( p, A ) ;
1-1-3°         q := Next( q, B ) ;
1-1-4°         r := Next( r, C ) ;
           }
1-2°       else if ( GreaterThan( Retrieve( p, A ), Retrieve( q, B ) ) )
1-2-1°         q := Next( q, B ) ;
1-2-2°       else   p := Next( p, A ) ;
           }
} /* Intersection */

```

上記の手続きにおいて, リスト  $A, B, C$  を配列で実現しても, 連結リストで実現しても, 関数  $First(\cdot), Makenull(\cdot), Tail(\cdot), Retrieve(\cdot, \cdot), Next(\cdot, \cdot)$  のいずれも  $O(1)$  で実行できる. 従って, 関数  $Same(x, y)$  および  $GreaterThan(p, q)$  がどちらも  $O(1)$  で実行できるならば, 1° の繰り返しにおける 1 回の操作に要する計算量は,  $O(1)$  であり, 1° 以外に要する計算量も  $O(1)$  である.

1° の繰り返しでは, その 1 回の操作において少なくとも  $p$  あるいは  $q$  の少なくとも一方は, 関数  $Next$  によって次の位置に進むから, 1° の繰り返しの総数は  $|A| + |B|$  を越えない. ここで,  $|A|$  および  $|B|$  はそれぞれ集合  $A$  および  $B$  の要素の個数である. 従って, 手続き  $Intersection$  の計算量は  $O(|A| + |B|)$  である.