

## 第 6 周作业答案及提示

### 1.

(1) Head(Tail(Tail(L1)))	(2) Head(Head(Tail(L2)))	(3)
Head(Head(Tail(Tail(Head(L3)))))		
(4) Head(Head(Tail(Tail(L4))))	(5) Head(Tail(Head(L5)))	(6 取消)
Head(Head(Tail(Head(Tail(L6)))))		

### 2.

```
(1) int akm(int m, int n) {
    if(m==0) return n+1;
    if(n==0) return(akm(m-1,1));
    return akm(m-1, akm(m,n-1));
}

(2) int akm_s(int m, int n){
    Stack s;
    s.push(m,n);
    do {
        while(m>0){
            while(n>0) {n--; s.push(m,n); }
            pop(m,n);
            m--;
            n=1;
            s.push(m,n);
        }
        s.pop(m,n);
        v=n+1;
        if(!s.IsEmpty()) {
            pop(m,n);
            m--;
            n=v;
            push(m,n);
        }
    }while(!s.IsEmpty());
    return v;
}
```

### 3.

分析:

安放第  $i$  行皇后时，需要在列的方向从 0 到  $n-1$  试探 ( $j = 0, \dots, n-1$ )。

在第  $j$  列安放一个皇后：

如果在列、主对角线、次对角线方向有其他皇后，则出现攻击，撤消在第  $j$  列安放的皇后。

如果没有出现攻击，在第  $j$  列安放的皇后不动，递归安放第  $i+1$  行皇后。

设置 4 个数组

$col[n]$  :  $col[i]$  标识第  $i$  列是否安放了皇后；

$md[2n-1]$  :  $md[k]$  标识第  $k$  条主对角线是否安放了皇后；

$sd[2n-1]$  :  $sd[k]$  标识第  $k$  条次对角线是否安放了皇后；

$q[n]$  :  $q[i]$  记录第  $i$  行皇后在第几列。

算法思路：

```
void Queen( int i ) {
    for ( int j = 0; j < n; j++ ) {
        if ( 第 i 行第 j 列没有攻击 ) {
            在第 i 行第 j 列安放皇后；
            if ( i == n-1 ) 输出一个布局；
        } else Queen ( i+1 );
        撤消第 i 行第 j 列的皇后；
    }
}
```

算法求精

```
void Queen( int i ) {
    for ( int j = 0; j < n; j++ ) {
        if ( !col[j] && !md[n+i-j-1] && !sd[i+j] )
        {
            /*第 i 行第 j 列没有攻击 */
            col[j] = md[n+i-j-1] = sd[i+j] = 1;
            q[i] = j;
            /*在第 i 行第 j 列安放皇后*/
            if ( i == n-1 ) {          /*输出一个布局*/
                for ( k = 0; k < n; k++ ) cout << q[k] << " ";
                cout << endl;
            }
            else Queen ( i+1 );
            col[j] = md[n+i-j-1] = sd[i+j] = 0;
            q[i] = 0;    /*撤消第 i 行第 j 列的皇后*/
        }
    }
}
```