Exercise 1.

(a) 基本思想：找出最底下的起始点 p_0（其必为凸包上的点），将其它点按与 p_0 的极角大小进行排序，从小到大遍历有序点集（保证逆时针方向遍历），利用栈后进先出的性质去除非凸包的顶点.

伪代码：1. 求出 Q 中 y 坐标值最小的点 p_0.

2. 按照与 p_0 的极角大小排序 Q 中其余点，结果为 $\langle p_1, \cdots, p_n \rangle$

3. Push p_0, p_1, p_2 into 栈 S.

4. For i=3 To n Do

5. While Top(S) 与 p_i 形成非左移动 Do

6. Pop(S)

7. Push(p_i, S)

8. Return.

(b) 时间复杂度. 第1行伪代码操作 $T_1(n) = O(n)$

第2行伪代码操作 $T_2(n) = O(n\log n)$

第3-7行伪代码操作 $T_3(n) = O(n)$

故 $T(n) = O(n\log n)$.

(c) 由循环不变量方法证明算法正确性.

循环初始：S 中 p_0, p_1, p_2 为 $\langle p_0, p_1, p_2 \rangle$ 点所构成的凸包

循环步骤：每次循环中，$\langle p_0, \cdots, p_i \rangle$ 中的非凸包点均被弹出 S，故 S 中的点构成 $\langle p_0, \cdots, p_i \rangle$ 点集的凸包

循环终止：循环终止时，S 中的点构成 $\langle p_0, \cdots, p_n \rangle$ 点集中的凸包，故算法正确.

Exercise 2

(a). 基本思想:

由公式 $a^n \% c = (a\%c)^n \% c$ (防止 a 过大导致溢出)

$$\begin{cases} a^n \% c = (a^{\frac{n}{2}})^2 \% c & n\text{为偶数} \\ a^n \% c = [(a^{\frac{n}{2}})^2 \cdot a] \% c & n\text{为奇数} \end{cases}$$

知,可设计分治算法,递归求解快速幂。

伪代码: Calculating (a, n, c)

1. if $n=1$ then return $a\%c$
2. else if $n\%2=0$
3. then return [Calculating $(a, \frac{n}{2}, c)$ · Calculating $(a, \frac{n}{2}, c)$] %c
4. else if $n\%2=1$
5. then return [Calculating $(a, \frac{n}{2}, c)$ · Calculating $(a, \frac{n}{2}, c)$ · $(a\%c)$] %c

(b). 时间复杂度 $T(n) = T(n/2) + O(1)$

$T(n) = O(\log n)$

Exercise 3

(a) 基本思想：多次利用中位数将问题划分为两个不同子问题，直至不可分为止（递归划分），再递归地进行合并，利用分治算法完成该问题

伪代码：1. 将n个建筑物的x轴坐标和高度按 l 由小到大排序，得 $(l_1', h_1', r_1') \cdots (l_n', h_n', r_n')$ 并放入集合 S_n，输出 (l_1', h_1') 和 $(r_n', 0)$ 输出左右边界

skyline(S, l, r, direction) 标记方向，返回值时用

1. if $l = r$ 不可分时的情况
2. then if direction = right
3. then return l 返回左边界点下标
4. else if direction = left
5. then return r 返回右边界点下标
6. else
7. $m =$ skyline$(S, l, \frac{l+r}{2}, left)$ 左右划分子问题
8. $n =$ skyline$(S, \frac{l+r}{2}, r, right)$
9. if $S_m.r < S_n.l$ 不相交情况的合并
10. then 输出 $(S_m.r, 0)$ 和 $(S_n.l, S_n.h)$
11. else 依次向右比较，直到 $S_n.r < S_n.r$ 相交情况的合并
12. if $S_m.r > S_n.r$ and $S_m.h \neq S_n.h$ 有两轮廓点情况
13. then 输出两个轮廓点（凸起部分）
14. else if $S_m.r < S_n.r$ and $S_m.h \neq S_n.h$ 一个轮廓点的情况
15. then 输出一个轮廓点
16. return direction == right ? l : r.

根据direction判定返回左边界下标还是右边界下标

(b) 时间复杂度

排序 $T_1(n) = O(n\log n)$

递归：$T_2(n) = 2 \cdot T(n/2) + O(n)$

$T_2(n) = O(n \cdot \log n)$

综上 $T(n) = O(n\log n)$

Exercise 4:

基本思想：将n个元素划分为m个非空集合，可分为两种情况：

①.将前面n-1个元素分为m-1个集合，最后一个元素单独一个集合

②.将前面n-1个元素分为m个集合，最后一个元素放入这些集合之一

由此可得式子 $N(n,m) = N(n-1,m-1) + m\,N(n-1,m)$，

故可用递归分治算法完成

伪代码：

```
Setnum (n, m)
if  m>n || m==0        错误情况
    then return 0
else if  m==n || m==1 || n==1      递归终止
    then return 1
else
    then return Setnum(n-1, m-1)+m*Setnum(n-1, m)
```