EX3 实验报告

王嵘晟　PB1711614

EX3-1

思路：这道题我对于输入的点坐标，先进行对横坐标由小到大排序，使用计数排序。然后对排序好的点，此题可以转化为类似最长上升子串的动态规划经典模型。使用自底向上法做动态规划，用 MaxLen[n] 记录前 n 个点中最长的上升子串长度。转移方程：

$$MaxLen[n] = \begin{cases} 1 \quad , \quad n = 1 \\ MAX_{1 \le i < n, A[i] < A[n]}(MaxLen[i] + 1, MaxLen[n]) \quad , \quad n > 1 \end{cases}$$

计数排序时间复杂度 $O(n)$，对 1 到 n 确定 MaxLen 的值由于双重循环时间复杂度 $O(n^2)$，遍历 MaxLen 找到需要输出的最长子串值时间复杂度 $O(n)$，所以总的时间复杂度为 $O(n^2)$

EX3-2

思路：这道题我开设了二维数组，m[i][j] 表示从第 i 个数组到第 j 个数组合并所需要的最小开销，状态转移方程：

$$m[i][j] = MIN(m[i][j], m[i][k] + m[k+1][j] + SUM(a, i, j+1)), i \le k \le j$$

由于这里使用了三重循环，第一重循环 k 为数组序列长度，i 为起始数组号，k 为序列中间指针，所以时间复杂度为 $O(n^3)$

EX3-3

思路：我开了个结构体来存储每一种物品的质量，价值和可选数量。多重背包我一开始想转化为部分背包来处理，使用贪心算法，结果发现了自己逻辑有错误，贪心算法

并不能行得通。于是尝试最普通的三重循环做动态规划，结果发现会超时。于是采用了二进制优化。对于任何数，一定可以找到一串 2 的幂次数的组合来表示。所以我用 cnt 来表示 $2^n, n = 0, 1, 2, 3...$，对于第二重循环 j 从背包最大承重 W 到每个物体的质量 w_i，j 每次减去的量为 $min(cnt.num_i) * w_i$。令 f[j] 表示背包中有最大序号为 i 的物品时最大价值。转移方程为：

$$f[j] = MAX(f[j], f[j - min(cnt.num_i) * w_i] + min(cnt.num_i) * v_i), 1 \leq i \leq j \leq n$$

时间复杂度为 $O(nlgW)$

EX3-EX

思路：这道题我用了一个比较巧妙的方法，把输入的 0 和 1 恰好取反，即 0 变 1，1 变 0，这样就可以用二维数组本身的元素进行计数，把 0 1 串读入二维数组 a，计数数组 Square 先做第一次计数————边长为 1 的正方形个数，然后根据状态方程：

$$Square[i][j] = \begin{cases} 0 \quad , \quad a[i][j] = 0 \\ MIN(Square[i-1][j-1], Square[i-1][j], Square[i][j-1]) + 1 \quad , \quad a[i][j] = 1 \end{cases}$$

求出数组 Square[i][j] 的每个元素大小后，做累加即为总的正方形个数，时间复杂度 $O(n^2)$