研究生算法课课堂笔记

上课日期：10.17　　　　　　　　第(2)节课

组长学号及姓名：1601214477 彭广举
组员学号及姓名：1601111289 马连韬
1601214484 王钰翔

一、滑雪问题

递归方法	1. 遍历整个数组，对每一个位置进行四个方向的移动判断，取使得滑雪长度最大的方向。 2. 如果该方向上的点的滑雪长度是未知的，则递归求得。	注意可能存在重复计算的问题
递推方法	1. 将整个数组排序 2. 定义辅助数组，用于记录每个数值的原始位置 3. 如果从最小的数值开始滑雪，则总滑雪长度为 1 4. 从低到高，依次计算每个点在四个方向上的滑雪长度，并取最大值	注意判断数组越界

二、最大连续子序列和问题

定义$f(j)$	以第 j 个元素结尾的子数组的和的最大值，即a_j元素肯定被选中
递推方程	$max(\ a_j,\ f(j-1)\ +\ a_j)$
初始条件	$f(1) = max(a1,\ 0)$
返回值	$max\ f(i),\ \ i = 1 \ldots n$
时间复杂度	$O(n)$
空间复杂度	在递推过程中，f 的计算仅依赖于前一个值，因此可以将空间复杂度将为常数级别

三、最大子矩阵和问题，是最大子序列和的变种

```
| a11 ...... a1i ......a1j ......a1n |
| a21 ...... a2i ......a2j ......a2n |
|  .    .    .    .    .    .    .  |
|  .    .    .    .    .    .    .  |
| ar1 ...... ari ......arj ......arn |
|  .    .    .    .    .    .    .  |
|  .    .    .    .    .    .    .  |
| ak1 ...... aki ......akj ......akn |
|  .    .    .    .    .    .    .  |
| an1 ...... ani ......anj ......ann |
```

思路	将二维情况降维至简单的一维问题
	1. 假设最终具有最大和的子矩阵为，第 r 行至第 k 行，第 i 列至第 j 列。 2. 当行的起始位置 r 与 k 确定时，只有列的起始位置 i 与 j 移动。（编者注：想象成拉手风琴。）如果某一列入选最大子矩阵，则它的第 r 行至第 k 行之间的元素都会被计算入最终的总和。

	3. 那么，可以把同一列上的全部元素求和，即将 r ~ k 行压缩为一行。则问题被简化为最大连续子序列和问题。一维序列为 $[a_{r1}+\ldots+a_{k1},\ \ldots,a_{ri}+\ldots+a_{ki},\ldots,\ a_{rj}+\ldots+a_{kj},\ldots,a_{rn}+\ldots+a_{kn}]$ 4. (1) 为此，可以先将第 1 行至第 2 行，第 1 行至第 3 行，第 1 行至 r 行，第 1 行至第 k 行，第 1 行至第 n 行对应列的所有元素求和。注意计算下一行的结果时应该直接利用上一行的结果，以免重复计算。 用第 1 行至第 k 行的结果减去第 1 行至第 r 行的结果，则获得第 r 行至第 k 行各个列的求和结果。这样即可避免重复计算。 (2) 另一种求和思路，同上计算第 1 行至第 2 行，第 1 行至第 3 行，第 1 行至第 n 行对应列的所有元素求和。随后用同样的方法计算第 2 行至第 3 行，第 2 行至第 4 行，第 2 行至第 n 行的求和结果。最终获得任意第 r 行到第 k 行的每列求和结果。
时间复杂度	用 O(n^2)的时间遍历起止行号，即枚举 n^2 种可能。每种可能下，用 O(n)的时间进行最大子序列和的计算。则总时间复杂度为 O(n^3)。 此外，需保证压缩的过程是线性的。
空间复杂度	根据求和方法不同，为平方级别或线性级别

四、题目：BUY LOW, BUY LOWER

题目大意：给定一个数组$\{a_1, a_2, \dots, a_j, \dots, a_n\}$，求不重复的最长下降子序列个数。

最长下降子序列问题：

$f(j)$	表示以第j个元素结尾的最长下降子序列的长度
递推方程	$f(j) = \max\limits_{\substack{0<i<j \\ a_i>a_j}} f(i) + 1$
初始值	$f(1) = 1$
返回值	$\max f(j)$

本题还需要统计个数：

$g(j)$	表示以第j个元素结尾的最长下降子序列的个数
递推方程	$g(j) = \sum\limits_{\substack{0<i<j \\ a_i>a_j \\ f(j)=f(i)+1}} g(i)$
初始值	$g(1) = 1$
返回值	$\sum\limits_{f(k)=max f(j)} g(k)$

由于数组中存在重复值，所以可能出现重复的最长子序列，导致重复计算。如果$f(i) == f(j)\ \&\&\ a_i == a_j$，则它们的下降子序列的位置是一样的，值也是一样的，只需要计算一次即可。

时间复杂度	$O(n^2)$用于求最长下降子序列，$O(n^2)$用于统计个数，总的时间复杂度为$O(n^2)$。
空间复杂度	需要保存f和g的中间结果，空间复杂度为$O(n)$。

五、问题：对于一个连续数组，求两个不相交的连续子序列，使其和最大。

Given a set of n integers: A={a1, a2,..., an}, we define a function d(A) as below:

$$d(A) = \max_{1 \le s_1 \le t_1 < s_2 \le t_2 \le n} \left\{ \sum_{i=s_1}^{t_1} a_i + \sum_{j=s_2}^{t_2} a_j \right\}$$

Your task is to calculate d(A).

原题：POJ 2479

对于长度为 n 的数组

定义 f(i)	为从 1 到 i 内的最大连续子序列和（即第一个子序列）
定义 g(i)	为从 i 到 n 段内的最大连续子序列和（即第二个子序列）

求出 f(i)及 g(i)，仅需枚举切割点的位置，即可得出最大两个连续子序列的和。

时间复杂度分析	求 f(j)和 g(j)为最大连续子序列和为题，时间复杂度为 O(n)。 枚举切割点的时间复杂度为 O(n)。故总的时间复杂度为 O(n)
空间复杂度分析	需要辅助数组保存 f(i)及 g(i)的值，因此空间复杂度为 O(n)

另外一种方法：

对于数组 v[1], v[2], ..., v[n]

定义 a(i)	前 i 个数的 1 个连续子串的最大值， 必须包括第 i 个	a(i) = max{a(i-1)+v[i], v[i]}, a(1)=v[1]
定义 b(i)	前 i 个数的 1 个连续子串的最大值， 不必包含第 i 个	b(i) = max{a(i), b(i-1)}, b(1)=v[1]
定义 c(i)	前 i 个数的 2 个连续子串的最大值， 必须包含第 i 个	c(i) = max{b(i)+v[i], c(i-1)+v[i]}, c(2) = v[1] + v[2]

return value	max c(i), I = 2, 3, …, n	
时间复杂度 和空间复杂度	均为 O(n)	