1. **问题描述**

现在有F束不同品种的花束，同时有至少同样数量的花瓶被按顺序摆成一行，其位置固定于架子上，并从1至V按从左到右顺序编号，V是花瓶的数目(F<=V)。花束可以移动，并且每束花用1至F的整数唯一标识，标识花束的整数决定了花束在花瓶中排列的顺序，如果i<j，花束i必须放在花束j左边的花瓶中。每个花瓶只能放一束花。如果花瓶的数目大于花束的数目，则多余的花瓶空置。

每一个花瓶都具有各自的特点。因此，当各个花瓶中放入不同的花束时，会产生不同的美学效果，并以一美学值(一个整数)来表示，空置花瓶的美学值为零。为取得最佳美学效果，必须在保持花束顺序的前提下，使花束的摆放取得最大的美学值。请求出具有最大美学值的一种摆放方式。

1. **问题分析**

该问题一个很重要的突破点在于“如果i<j，花束i必须放在花束j左边的花瓶中”，因而可以借用该性质。

不放假设二维数组dp[F][V]，对于其中的每个元素dp[i][j]，其代表的意义为第i朵花放入编号小于等于j的花瓶中所能取得的最大美学值。

则对于任意问题dp[i][j]，可划分为两个子问题：

dp[i][j-1]，即第i朵花不放入第j个花瓶，故其值等于第i朵花放入编号小于等于j-1的花瓶。

dp[i-1][j-1]+beauty[i][j]，即第i朵花放入第j个花瓶，但此时第i-1朵花不再能放入第j个花瓶。

且最小子问题为dp[0][0~V]，其值均为0，即不放入花时，美学值均为0。

1. **输入输出格式**

输入：F（花束个数）、V（花瓶个数）

输出：最大美学值、花束编号从1~F应放入的花瓶编号

1. **基本算法**
2. 算法思想

由问题分析可知，线性规划的递推关系式为

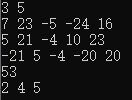
因而构造一个dp[F+1][V+1]大小的二维矩阵，其中第一行全置为0。从第二行开始，从j=1向j=V按照递推关系式计算即可。

1. 测试情况



在编写完成并简单测试后搜索得知，该问题为POJ 1157原题，遂在POJ上验证了样例，并提交了代码进行评测。





1. **总结**

通过本次作业，我对利用线性规划解决问题有了更深入的认识。线性规划的重点在于寻找递推关系式，分析如何划分子问题并全遍历，这次实验更多的让我学会了后者。

这个问题其实一开始想了很久，觉得无论如何都需要全遍历所有排列，但是经过仔细读题后发现“如果i<j，花束i必须放在花束j左边的花瓶中”，因而实际上该问题的“全遍历”并不是花束摆放的“全排列”，这给我设计dp的二维数组时，将列视为“放入编号小于等于j的花瓶”提供了关键性的帮助。

通过该算法的学习与设计分析，我对动态规划有了更深入的认识，让我在设计动态规划算法时想到的不仅仅是递推关系式，还需要考虑问题的“全遍历”到底是什么。

非常感谢老师的作业设计！