### 花束摆放问题

1. 问题描述

现在有束不同品种的花束，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至按从左至右顺序编号，是花瓶的数目（)。花束可以移动，并且每束花用1至的整数唯一标识。标识花束的整数决定了花束在花瓶中的顺序，如果，花束必须放在花束左边的花瓶中，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)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. 问题分析与设计思路

令为第束花在第个花瓶中时，前束花最大的美学值

则递推公式为：



1. 代码实现

时间复杂度：

1. **void** solution(){
2. **int** fNum=3;
3. **int** vNum=5;
4. **int** value[3][5]={7,23,-5,-24,16,
5. 5,21,-4,10,23,
6. -21,5,-4,-20,20};
7. **int** dp[3][5];
8. **int** len=vNum-fNum;
9. **for**(**int** i=0;i<fNum;i++)
10. **for**(**int** j=0;j<vNum;j++)
11. dp[i][j] =0;
13. **for**(**int** i=0;i<=len;i++)
14. dp[0][i]=value[0][i];
15. **for**(**int** i=1;i<fNum;i++){
16. **for**(**int** j=i;j<=len+i;j++){
17. dp[i][j]=dp[i-1][i-1]+value[i][j];
18. **for**(**int** k=i;k<j&&k<=len+i-1;k++){
19. dp[i][j] =max(dp[i][j],dp[i-1][k]+value[i][j]);
20. }
21. }
22. }
23. **for**(**int** i=0;i<fNum;i++){
24. **for**(**int** j=0;j<vNum;j++)
25. cout<<dp[i][j]<<" ";
26. cout<<endl;
27. }
28. }
29. 结果

输入：value表，行为花束，列为花瓶

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | 7 | 23 | -5 | -24 | 16 |
| 2 | 5 | 21 | -4 | 10 | 23 |
| 3 | -21 | 5 | -4 | -20 | 20 |

输出：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | 7 | 23 | -5 | 0 | 0 |
| 2 | 0 | 28 | 19 | 33 | 0 |
| 3 | 0 | 0 | 24 | 8 | 53 |

可见53是最大美学值，这也可以直接从value表中直接看出