

## HDOJ 1024 Max Sum Plus Plus

作者：admin 发布：2013-03-23 21:03 分类：Algorithm, HDOJ, 动态规划 阅读：144 浏览数 抢沙发

描述：

题意的大致意思就是求解数组中的最大m段和。

思路：

利用动态规划思想， $dp[i,j]$ 表示前j项中分为i段的和的最大值，由此推导出 $dp[i,j]=\max\{dp[i,j-1]+Num[j], \max\{dp[i-1,t]+Num[j]\}, \text{其中 } i-1 \leq t < j\}$ ;

情况一分析： $dp[i,j-1]+Num[j]$ ，意思就是 $Num[j]$ 包含在i段之中的。

情况二分析： $\max\{dp[i-1,t]+Num[j]\}$ ，意思就是 $Num[j]$ 独立划分成为一段。

优化处理：

观察上述两种情况可得 $dp[i,j]$ 的值只和 $dp[i,j-1]$ 和 $\max\{dp[i-1,t]\}$ ，其中 $i-1 \leq t < j$ 这两个值相关，于是我们可以用 $dp[j]$ 表示现阶段的最大值，用 $B[j]$ 表示上一阶段的最大值即表示 $\max\{dp[i-1,t]\}$ 。

实现代码：

```
01 #include "stdio.h"
02 #define INF 99999999
03 #define N 1000010
04
05 int Num[N], dp[N], B[N];
06
07 int main()
08 {
09     int n, m, i, j, max;
10     while (scanf("%d%d", &m, &n) == 2)
11     {
12         for (i = 1; i <= n; i++)
13         {
14             scanf("%d", &Num[i]);
15         }
16
17         //初始化
18         for (i = 0; i <= n; i++)
19         {
20             dp[i] = 0;
21             B[i] = 0;
22         }
23
24         for (i = 1; i <= m; i++)
25         {
26             max = -INF;
27
28             for (j = i; j <= n; j++)
29             {
30                 //这里就是动态转移方程的实现
31                 if (dp[j-1] > B[j-1])
32                 {
33                     dp[j] = dp[j-1] + Num[j];
34                 }
35                 else
36                 {
37                     dp[j] = B[j-1] + Num[j];
38                 }
39
40                 B[j-1] = max; //存放dp[]阶段的相应最大值
41             }
42         }
43     }
```

```
42         if(max<dp[j])
43         {
44             //应为dp[j]即将成为B[]数组中的值，而B[]数组中存放的是上一阶段的最大值
45             //现在不断更新max值，循环结束后存放在B[]数组中
46             max=dp[j];
47         }
48     }
49 }
50
51 B[j-1]=max;//最后一个阶段位置的初始化，即为dp[]阶段的最大值
52 }
53
54 printf("%d\n",max);
55
56 }
57
58 return 0;
59 }
```

本文固定链接: <http://www.icrany.com/98.html> | iCrany