

Chino的成绩

一. 考察内容:

动态规划

二. 题目分析:

[题目大意]

n 个物品一个容量为 m 的背包，每个物品体积为1，第 i 个物品连续装 j 个时可以多获得 $v[i][j]$ 贡献，求一种装物品的顺序，最大化贡献。

[写题思路]

考虑对 $v[i][j]$ 求前缀和，那么 $v[i][j]$ 的含义变为第 i 个物品连续装 j 个的总贡献。

考虑动态规划求解，设 $f[i][j][k]$ 为容量为 i 时最后装了 k 个 j 物品的最大贡献，显然等于装了 k 个 j 物品之前的最大贡献加上 $v[j][k]$ 。考虑用 $maxv[i][0]$ 储存容量为 i 时的最大贡献， $maxv[i][1]$ 储存容量为 i 时的次大贡献， $maxid[i]$ 储存容量为 i 时最大贡献的最后一个物品，那么如果 $maxid=i$ ，则 $f[i][j][k]=maxv[i][1]+v[j][k]$ ，否则 $f[i][j][k]=maxv[i][0]+v[j][k]$ 。 $maxv$ 和 $maxid$ 由 f 转移。

由于空间限制，我们可以考虑把 f 数组的第一维省去。

三. 代码实现:

```
#define _CRT_SECURE_NO_DEPRECATE
/*****
*创建时间: 2018 08 18
*文件类型: 源代码文件
*题目来源: 洛谷
*当前状态: 已通过
*备忘录: 动态规划
*作者: HtBest
*****/
#include <stdio.h>
#include <iostream>
#include <string>
#include <stdlib.h>
#include <string.h>
#include <algorithm>
#include <queue>
using namespace std;
#define MAXN 10001
#define MAXM 101
int n,m,maxid[MAXN]; //这里的n和m与题目相反
long long f[MAXN][MAXM],maxv[MAXM][2],a[MAXN][MAXM];
/* Variable explain:
f: 状态
maxv: 前i天的最大(0)/次大放法(1)
maxid: 最大放法最后放的东西
*/
void read()
{
    // freopen(".in","r",stdin);
    // freopen(".out","w",stdout);
    scanf("%d%d",&m,&n);
    for(int i=1;i<=n;++i) for(int j=1;j<=m;++j) scanf("%lld",&a[i][j]),a[i][j]+=a[i][j-1];
    return;
}
```

```
void dp()
{
    //状态f[i][j][k]前i天, 最后放了k个j的最大贡献
    for(int i=1;i<=m;++i)
    {
        for(int j=1;j<=n;++j)
        {
            for(int k=1;k<=i;++k)
            {
                f[j][k]=maxv[i-k][maxid[i-k]==j?1:0]+a[j][k];
                //后面k个不动, 前面i-k个取最大的放法
                // printf("f[%d][%d][%d]=%lld+%d\n",i,j,k,maxv[i-k][maxid[i-k]==j?
1:0],a[j][k]);
                if(f[j][k]>maxv[i][0])
                {
                    maxv[i][1]=maxv[i][0];
                    maxv[i][0]=f[j][k];
                    maxid[i]=j;
                }
                else maxv[i][1]=max(maxv[i][1],f[j][k]);
            }
            // printf("%d:%d %lld\n",i,maxid[i],maxv[i][0]);
        }
        printf("%lld\n",maxv[m][0]);
    }
}

int main()
{
    read();
    dp();
    return 0;
}
```

[<题目跳转>](#) [<查看代码>](#)