

## 1266: 机器分配

## 题目描述

总公司拥有高效设备M台，准备分给下属的N个分公司。各分公司若获得这些设备，可以为国家提供一定的盈利。问：如何分配这M台设备才能使国家得到的盈利最大？求出最大盈利值。其中 $M \leq 15$ ， $N \leq 10$ 。分配原则：每个公司有权获得任意数目的设备，但总台数不超过设备数M。

## 输入

第一行有两个数，第一个数是分公司数N，第二个数是设备台数M；

接下来是一个 $N \times M$ 的矩阵，表明了第 I 个公司分配 J 台机器的盈利。

## 输出

第一行输出最大盈利值；

接下N行，每行有2个数，即分公司编号和该分公司获得设备台数。

## 输入样例

```
3 3          //3个分公司分3台机器
30 40 50
20 30 50
20 25 30
```

## 输出样例

```
70           //最大盈利值为70
1 1          //第一分公司分1台
2 1          //第二分公司分1台
3 1          //第三分公司分1台
```

## 解析

本题其实是一个分组背包。你可以这么考虑：对于每一个公司，我们可以分配1台、2台到N台，共N种情况，那么这N种情况必然是彼此互斥的，所以我们可以把这N种情况看成一组，每种情况就是一个物品，所占的重量就是它的数量，这样就完美的切合了分组背包。

## 编码

```
#include<bits/stdc++.h>
```

```

using namespace std;

int bagV, m;

int v[1005][1005];    //第i个公司，分到j台机器的价值
int f[1005][1005];    //前i个公司，总数量为j台机器时的最大价值

//打印每个公司分配的数量
void show(int n, int m) {
    //不存在小于1的公司，直接返回
    if (!n) return;
    //从最后一组开始倒推，计算出每个公司实际使用的数量
    //f[i][j] = f[i-1][j] + v[i][j]
    //注意v[i][j]可以等于0;
    for (int j = 0; j <= m; j++) {
        if (f[n - 1][j] + v[n][m - j] == f[n][m]) {
            show(n - 1, j);
            //打印公司序号以及物品数量
            cout << n << " " << m - j << endl;
            return;
        }
    }
}

int main() {

    //公司为分组数，机器总数为背包的上限；
    cin >> m >> bagV;

    for (int i = 1; i <= m; i++) {
        for (int j = 1; j <= bagV; j++) {
            //获取输入的价值
            cin >> v[i][j];
        }
    }

    //遍历每个分组
    for (int k = 1; k <= m; k++) {
        //一维优化，分组背包本质也是01背包，因此从后向前滚动容量
        for (int j = bagV; j >= 1; j--) {
            //遍历每一组的数据
            for (int a = 1; a <= bagV; a++) {
                //取出组内的每一个物品
                if (j >= a) {
                    //组内比较
                    f[k][j] = max(f[k][j], f[k - 1][j - a] + v[k][a]);
                }
            }
        }
    }
}

```

```
        //与上一组比较
        f[k][j] = max(f[k][j], f[k - 1][j]);
    }
}

//最大值在最后一个背包
cout << f[m][bagV] << endl;

show(m, bagV);

return 0;
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

