

1294: Charm Bracelet

题目描述

经典0—1背包问题,有 n 个物品,编号为 i 的物品的重量为 $w[i]$,价值为 $c[i]$,现在要从这些物品中选一些物品装到一个容量为 m 的背包中,使得背包内物体在总重量不超过 m 的前提下价值尽量大。

输入

第1行:两个整数, n (物品数量, $n \leq 3500$)和 m (背包容量, $m \leq 12880$)。

第2.. $n+1$ 行: : 每行二个整数 $w[i]$, $c[i]$,表示每个物品的重量和价值。

输出

仅一行,一个数,表示最大总价值。

输入样例

```
4 6
1 4
2 6
3 12
2 7
```

输出样例

```
23
```

解析

01背包模板题,直接套用。

编码

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

using namespace std;

//第1行:两个整数, n (物品数量, n≤3500) 和m (背包容量, m≤12880)。
int bagV, n;

int w[3501];           //商品的体积
int v[3501];           //商品的价值

int dp[12881] = {0};   //动态规划表
```

```

int main() {

    //记录最大承重和物品数量
    cin >> n >> bagV;
    //记录每个物品的重量和价值
    for (int i = 1; i <= n; i++) {
        cin >> w[i] >> v[i];
    }

    //从放入第一件物品开始
    for (int i = 1; i <= n; i++) {
        //从第一个格子开始尝试
        for (int j = bagV; j >= 1; j--) {
            //如果当前的格子的重量小于目标物品的重量，则价值等于前一个物品的价值
            if (j < w[i])
                dp[j] = dp[j];
            else
                dp[j] = max(dp[j], dp[j - w[i]] + v[i]);
        }
    }

    //01背包的最大值在最后一个格子中
    cout << dp[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。

