

Oj.1013 无限背包问题

2019年2月4日 星期一 上午12:21

1013. 无限背包

Description

你现在有一个体积为 V 的大袋子，有 N 种物品，假设每种物品的数量有无限多个，而且第 i 种物品的体积是 $c[i]$,价值是 $w[i]$,请选择一些物品放入袋中，使袋中物品的价值总和最大。

注意每种物品的数量是无限多的；对于放入袋中的同种物品数量没有限制。

Input Format

第一行包含两个正整数 V 和 N ，分别代表袋子的体积和物品的种类数。

以下 N 行分别由2个正整数组成，代表每种物品的体积和价值。

$V \leq 10000, N \leq 1000$

保证操作可在C++ int范围内完成。

Output Format

输出一个整数，表示最大的价值总和

Sample Input

```
5 3
2 3
3 2
4 1
```

Sample Output

```
6
```

实在不知道怎么做

就网上查了一下

Dp

也不算dp

说不清楚

Sum[j]表示体积小于v时的放的物品的最大价值

即sum[bagVolumn]为答案

刚开始把所有0号物品放进去

计算sum[0] 到sum[bagVolumn]

看能到多大

然后放1号

从放一个到放多个

如果能比原来大，就替换

不能

就用原来的

直到计算了所有种类的物品

```
for(i = 0; i < boxNum; i++){  
    for(j = volumn[i]; j <= bagVol; j++){  
        sum[j] = max(sum[j], sum[j - volumn[i]] + value[i]);  
    }  
}
```