

{% note info %} **摘要** Title: 5. 多重背包问题 II Tag: 多重背包 Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by: NEFU AB-IN

[Link](#)

@TOC

5. 多重背包问题 II

- 题意

有 N 种物品和一个容量是 V 的背包。第 i 种物品最多有 s_i 件，每件体积是 v_i ，价值是 w_i 。求解将哪些物品装入背包，可使物品体积总和不超过背包容量，且价值总和最大。输出最大价值。

$0 < N \leq 1000$ $0 < V \leq 2000$ $0 < v_i, w_i, s_i \leq 2000$

- 思路

多重背包利用**二进制倍增**优化

比如第一种物品，有200件，我们没有必要将所有的都枚举一遍，可以对其进行**分组** 1, 2, 4, 8, 16, 32, 64, 73 1, 1, 1, 1,, 1,, 1,, 1,, 1 第一行的数，对应第二行的二进制串，0代表不取，1代表取，即每个数只取一次（其实就符合了**01背包**的前提），这样就可以将1~200的所有数全表示出来，比如这个二进制串就代表了取第一种物品200件

那么第二种、第三种、...物品就都可以用这种方式来表示，用新的数组做01背包就可以了

- 代码

```
...
Author: NEFU AB-IN
Date: 2022-03-06 14:39:14
FilePath: \ACM\Acwing\5.py
LastEditTime: 2022-03-06 14:47:29
...

N = int(2e5 + 10) #NlogS上取整
w, v, dp = [0] * N, [0] * N, [0] * N

n, m = map(int, input().split())

cnt = 0
for i in range(n):
    a, b, s = map(int, input().split())
    k = 1
    while s - k >= 0:
        cnt += 1
        v[cnt] = k * a
```

```
w[cnt] = k * b
s -= k
k *= 2
if s > 0: #若有剩余
    cnt += 1
    v[cnt] = s * a
    w[cnt] = s * b

n = cnt
for i in range(1, n + 1):
    for j in range(m, v[i] - 1, -1):
        dp[j] = max(dp[j], dp[j - v[i]] + w[i])

print(dp[m])
```