

O题题解报告

题意：多重背包问题

一看就是多重背包（虽然当时不知道）

背包容量就是最大重要度之和W

对于多重背包，我们可以把它转化成01背包问题

当然不能有几重背包就转化成几个背包

利用一个数学结论进行优化（神奇的二进制）

任何一个数都能用2的1到n次方的组合来表示

我们只要把每个多重背包拆成log级别的个数就行了

```
while(cnt<m){  
    val[++tot]=cnt*v;  
    weight[tot]=cnt*c;  
    m-=cnt;  
    cnt=cnt<<1;  
}
```

比如某种宝可梦有8个，每个价格是3，

那么我们就可以拆成1个价格为 3×1 的宝可梦；

1个价格为 3×2 的宝可梦；

1个价格为 3×4 的宝可梦；

1个价格为 $3 \times (8-4-2-1)$ 的宝可梦；

挑选该种1~8个宝可梦都可以通过上述四个背包来实现。

因为个数很少的原因，这里拆法的优秀体现的不是很充分，当个数很多的时候就能体现出来了。

时间复杂度 $O((n \log m)^2)$:

每个宝可梦都可以拆成 $\log m$ 个左右的背包，一共有 $n \log m$ 个背包，套入01背包时间复杂度就能得出多重背包的复杂度