

# 多重背包、混合背包

前置知识:

讲解054 - 单调队列

讲解067、讲解068 - 二维动态规划及其空间压缩技巧

讲解073 - 01背包

讲解074 - 完全背包

【必备】课程的动态规划大专题从讲解066开始，建议从头开始学习会比较系统

多重背包：每一种物品给定数量的限制，进行可能性展开

多重背包的枚举优化：二进制分组优化（最常用）、单调队列优化（复杂度最好，理解稍难）

混合背包：多种背包模型的组合与转化

注意：

动态规划优化枚举是一个很大的话题，这节课讲了二进制分组优化、单调栈优化

动态规划优化枚举的技巧，在 讲解082 - 动态规划优化枚举的技巧 会有进一步的讲解

【扩展】课程阶段 也会有进一步的讲述

# 多重背包、混合背包

## 题目1

多重背包不进行枚举优化

宝物筛选

一共有 $n$ 种货物, 背包容量为 $t$

每种货物的价值( $v[i]$ )、重量( $w[i]$ )、数量( $c[i]$ )都给出

请返回选择货物不超过背包容量的情况下, 能得到的最大的价值

测试链接 : <https://www.luogu.com.cn/problem/P1776>

# 多重背包、混合背包

## 题目2

多重背包通过二进制分组转化成01背包(模版)

### 宝物筛选

一共有 $n$ 种货物, 背包容量为 $t$

每种货物的价值( $v[i]$ )、重量( $w[i]$ )、数量( $c[i]$ )都给出

请返回选择货物不超过背包容量的情况下, 能得到的最大的价值

测试链接: <https://www.luogu.com.cn/problem/P1776>

核心在于:

可能有一些张数情况有重复计算, 但是不会漏掉任何一种张数情况, 也不会多算任何一种张数情况

因为是二进制分组, 让原本 $O \sim cnt$ 规模的枚举, 变成了 $O \sim \log(cnt)$ 规模的枚举

```
for (int k = 1; k <= cnt; k <= 1) {  
    v[++m] = k * value;  
    w[m] = k * weight;  
    cnt -= k;  
}  
if (cnt > 0) {  
    v[++m] = cnt * value;  
    w[m] = cnt * weight;  
}
```

# 多重背包、混合背包

## 题目3

观赏樱花

给定一个背包的容量 $t$ ，一共有 $n$ 种货物，并且给定每种货物的信息

花费( $cost$ )、价值( $val$ )、数量( $cnt$ )

如果 $cnt == 0$ ，代表这种货物可以无限选择

如果 $cnt > 0$ ，那么 $cnt$ 代表这种货物的数量

挑选货物的总容量在不超过 $t$ 的情况下，返回能得到的最大价值

背包容量不超过1000，每一种货物的花费都 $> 0$

测试链接：<https://www.luogu.com.cn/problem/P1833>

完全背包转化为多重背包，再把多重背包通过二进制分组转化为01背包

# 多重背包、混合背包

## 题目4

### 多重背包单调队列优化

#### 宝物筛选

一共有 $n$ 种货物, 背包容量为 $t$

每种货物的价值( $v[i]$ )、重量( $w[i]$ )、数量( $c[i]$ )都给出

请返回选择货物不超过背包容量的情况下, 能得到的最大的价值

测试链接 : <https://www.luogu.com.cn/problem/P1776>

注意:

需要理解 **讲解054 - 单调队列**, 不然听不懂

理解难度稍微大一些, 一定要配合具体例子来分析

# 多重背包、混合背包

## 题目5

混合背包 + 多重背包普通窗口优化

能成功找零的钱数种类

每一种货币都给定面值 $val[i]$ ，和拥有的数量 $cnt[i]$

想知道目前拥有的货币，在钱数为 $1、2、3...m$ 时

能找零成功的钱数有多少

也就是说当钱数的范围是 $1\sim m$

返回这个范围上有多少可以找零成功的钱数

比如只有3元的货币，数量是5张

$m = 10$

那么在 $1\sim 10$ 范围上，只有钱数是3、6、9时，可以成功找零

所以返回3表示有3种钱数可以找零成功

测试链接：<http://poj.org/problem?id=1742>