

01背包、有依赖的背包

前置知识：

讲解067、讲解068 - 二维动态规划及其空间压缩技巧

【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

本节课讲述：

01背包：每个物品 要和不要 两种可能性展开

有依赖的背包：多个物品 变成一个复合物品（互斥），每件复合物品 不要和怎么要 多种可能性展开

时间复杂度 $O(\text{物品个数} * \text{背包容量})$ ，额外空间复杂度 $O(\text{背包容量})$

不能用01背包来解，但是非常重要的问题：非负数组前k个最小的子序列和问题

注意：

讲解069已经讲了多维费用背包

01背包、有依赖的背包

题目1

01背包(模版)

给定一个正数 t ，表示背包的容量

有 n 个货物，每个货物可以选择1次

每个货物有自己体积 $costs[i]$ 和价值 $values[i]$

返回在不超过总容量的情况下，怎么挑选货物能达到价值最大

返回最大的价值

测试链接：<https://www.luogu.com.cn/problem/P1048>

01背包、有依赖的背包

题目2

夏季特惠

某公司游戏平台的夏季特惠开始了，你决定入手一些游戏

现在你一共有X元的预算，平台上所有的 n 个游戏均有折扣

标号为 i 的游戏的原价 a_i 元，现价只要 b_i 元

也就是说该游戏可以优惠 $a_i - b_i$ ，并且你购买该游戏能获得快乐值为 w_i

由于优惠的存在，你可能做出一些冲动消费导致最终买游戏的总费用超过预算

只要满足：**获得的总优惠金额 \geq 超过预算的总金额**

那在心理上就不会觉得吃亏

现在你希望在心理上不觉得吃亏的前提下，获得尽可能多的快乐值。

测试链接：<https://leetcode.cn/problems/tJau2o/>

01背包、有依赖的背包

题目3

目标和

给你一个非负整数数组 `nums` 和一个整数 `target` 。

向数组中的每个整数前添加 '+' 或 '-'，然后串联起所有整数

可以构造一个表达式

例如`nums=[2, 1]`，可以在2之前添加 '+'，在1之前添加 '-'

然后串联起来得到表达式 `"+2-1"`。

返回可以通过上述方法构造的，运算结果等于 `target` 的不同表达式的数目

测试链接：<https://leetcode.cn/problems/target-sum/>

01背包、有依赖的背包

题目4

最后一块石头的重量 II

有一堆石头，用整数数组 `stones` 表示，其中 `stones[i]` 表示第 `i` 块石头的重量

每一回合，从中选出任意两块石头，然后将它们一起粉碎

假设石头的重量分别为 `x` 和 `y`，且 $x \leq y$

那么粉碎的可能结果如下：

如果 $x == y$ ，那么两块石头都会被完全粉碎；

如果 $x \neq y$ ，那么重量为 `x` 的石头将会完全粉碎，而重量为 `y` 的石头新重量为 `y-x`

最后，最多只会剩下一块 石头，返回此石头 最小的可能重量

如果没有石头剩下，就返回 0

测试链接：<https://leetcode.cn/problems/last-stone-weight-ii/>

01背包、有依赖的背包

题目5

有依赖的背包(模版)

物品分为两大类：主件和附件

主件购买没有限制，钱够就可以；附件购买有限制，该附件所归属的主件先购买，才能购买这个附件

例如，若想买打印机或扫描仪这样的附件，必须先购买电脑这个主件。以下是一些主件及其附件的展示：

电脑：打印机，扫描仪 | 书柜：图书 | 书桌：台灯，文具 | 工作椅：无附件

每个主件最多有2个附件，并且附件不会再有附件，主件购买后，怎么去选择归属附件完全随意，钱够就可以

所有的物品编号都在1~m之间，每个物品有三个信息：价格v、重要度p、归属q

价格就是花费，价格 * 重要度 就是收益，归属就是该商品是依附于哪个编号的主件

比如一件商品信息为[300,2,6]，花费300，收益600，该商品是6号主件商品的附件

再比如一件商品信息[100,4,0]，花费100，收益400，该商品自身是主件($q==0$)

给定m件商品的信息，给定总钱数n，返回在不违反购买规则的情况下最大的收益

测试链接：<https://www.luogu.com.cn/problem/P1064>

测试链接：<https://www.nowcoder.com/practice/f9c6f980eeec43ef85be20755ddbeaf4>

01背包、有依赖的背包

题目6

非负数组前k个最小的子序列累加和

给定一个数组nums，含有n个数字，都是非负数

给定一个正数k，返回所有子序列中累加和最小的前k个累加和

子序列是包含空集的

$1 \leq n \leq 10^5$

$1 \leq \text{nums}[i] \leq 10^6$

$1 \leq k \leq 10^5$

注意这个数据量，用01背包的解法是不行的，时间复杂度太高了

对数器验证

时间复杂度 $O(n \cdot \log n) + O(k \cdot \log k)$ ，额外空间复杂度 $O(k)$

讲解025 ~ 讲解027 - 堆