

动态规划中的背包问题

原创 万文俊 LOA算法学习笔记 2021-01-09 22:45

假如你是一个盗墓贼（当然小伙伴们必须要遵纪守法），你来到了一个山洞，恰好发现了一堆宝藏，但是苦于你的背包是有限的，不能一次带走所有宝藏，聪明的你一定会想办法使得自己装的宝贝越值钱越好，这个办法正是研究的内容。

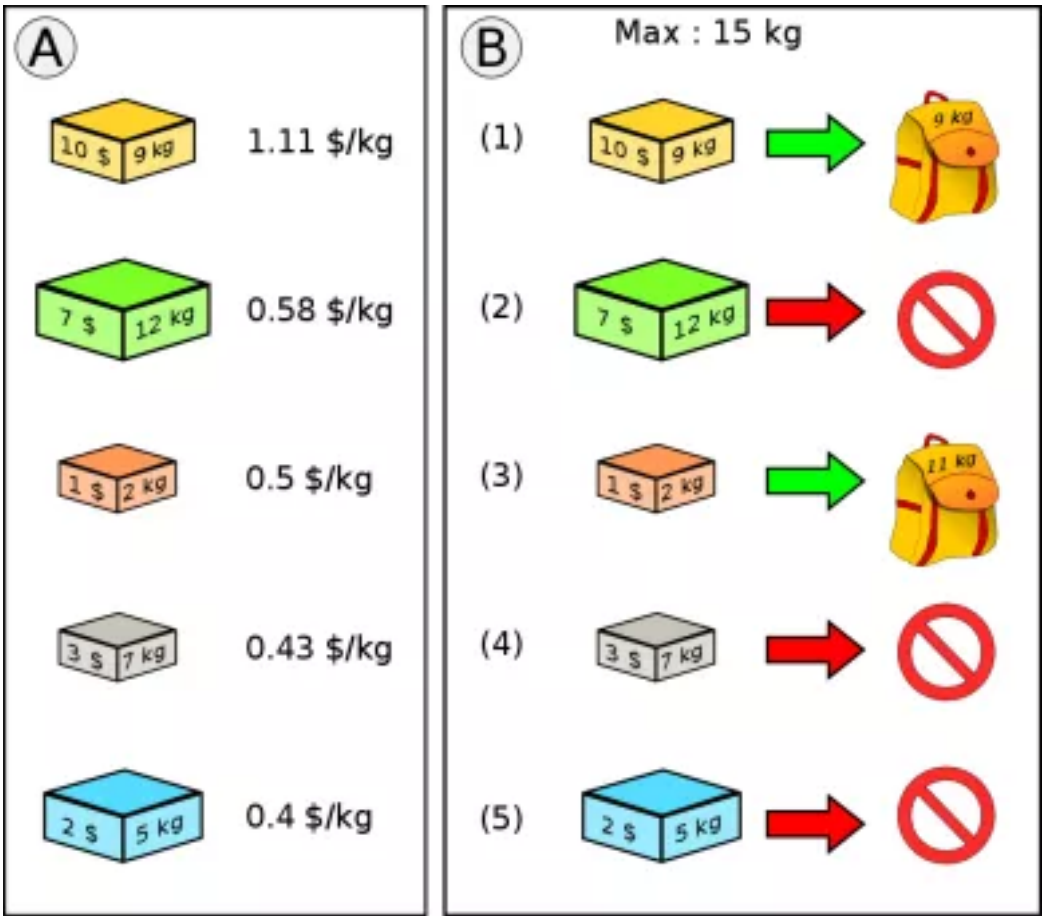
上述情景一般被称作**背包问题**，其中0-1背包问题最为经典，不妨从该问题说起。

01 “0-1”背包问题

问题描述：一个物品序列N，每个物品数量为1，重量 w_i ，价值 v_i ，背包大小为W。

问：如何使得背包里的货物价值最大

思考：第一反应往往是先挑选单位重量价值最高的物品，如图所示。



但这样存在一个问题，就是物品的重量信息被丢失了，从而不知道加入的物品是否超重。从答案也可以看出不是最优解。

此时盗墓贼陷入了困境，不知从何下手。

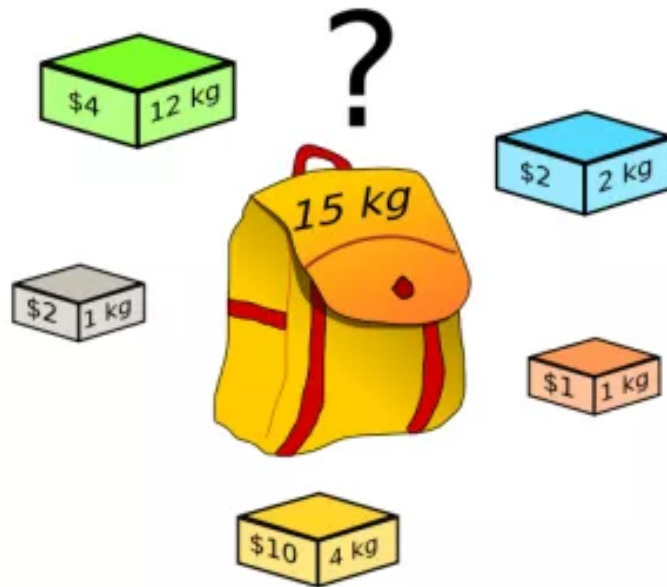
那不如**从最简单的case入手**：

N=1: 当你眼前只有一个物品时，那显然只要是满足其重量小于背包重量就拿走,会做。

N=2：当你眼前有两个物品时，有可能都拿走，也有可能都不拿走，或者只能拿一个，如果只能拿一个那肯定拿价值高的那个，这个也会做。

N=5：我们先给物品排个序：{物品1 (12kg|\$4)、物品2 (1kg|\$1)、物品3 (10kg|\$4)、物品4 (1kg|\$1)、物品5 (2kg|\$2)}

这个就比较困难，但也有办法，试想，一个物品只有加或者不加两种状态， 2^5 种组合方式，看起来也不是很复杂，勉强算是会做吧，能不能更简便一点呢？



不妨换一种思路，**不去想组合方式，而是首先观察原问题看起来像不像一个多步决策过程**，在第*i*步的决策时只考虑第*i*件物品选或不选，那么在做第*i*个决策的时候分两种情况：

- 1) 选：这样得到了第*i*个物品的价值，但是背包的容量减少了第*i*个物品的重量
- 2) 不选：这样背包的容量不变，但是价值也不变

一开始从第五个开始（这个是任意的，第一个或第五个都OK），当第五个物品选了的时候背包的容量变为15kg-2kg，其价值变为\$2；若是不选，则背包容量还是15kg，价值\$0，并且下次只能从第四个开始。

于是，在这样的多步决策过程中，原问题的最优解拆分成子问题的最优解的组合。我们来形式化一下这个过程：

- Step1：对物品进行排序（并不需要实际排序，只是处理问题时按照顺序进行）；
- Step2：最优子结构 $OPT[i][j]$ ，表示剩余空间为*j*时第*i*个物品的最优解，从第*i*个物品开始，每个物品只有两种状态，选或不选
- Step3：递推关系式 $OPT[i][j] = \max\{OPT[i-1][j], OPT[i-1][j-w_i] + v_i\}$ s.t. $j-w_i \geq 0$

02 完全背包问题

上述是背包问题的最简单的情形，让我们再推广一下，假如你发现洞穴里的财宝无穷无尽，每个宝贝都有无穷多个，可惜的是你的背包仍然是有限的，那么这个时候怎么才能让你的财富最大化呢？

问题描述：一个物品序列*N*，每个物品数量无限，重量 w_i ，价值 v_i ，背包大小为*W*。

问：如何使得背包里的货物价值最大。

思考：与0-1背包的区别就是物品的数量无限，从挑或不挑变成了挑几个（0也算），这样的话解法大同小异，其形式化过程如下：

- Step1：对物品进行排序；
- Step2：最优子结构 $OPT[i][j]$ ，表示剩余空间为*j*时第*i*个物品的最优解，从第*i*个物品开始，每个物品可以选*k*个
- Step3：递推关系式 $OPT[i][j] = \max\{OPT[i-1][j], OPT[i-1][j-k*w_i] + k*v_i\}$ s.t. $j-k*w_i \geq 0$ ，由于*k*是可以为0的，所以式子可以写成 $OPT[i][j] = \max\{OPT[i-1][j-k*w_i] + k*v_i\}$ s.t. $j-k*w_i \geq 0$

03 多重背包问题

多重背包问题就是每个宝贝不止一个，假如是 m_i 个。有了0-1背包和完全背包的基础，想必多重背包的问题也不难解决，

问题描述：一个物品序列*N*，每个物品数量 m_i ，重量 w_i ，价值 v_i ，背包大小为*W*。

问：如何使得背包里的货物价值最大

思考：只需要将限定条件稍加修改即可，其形式化过程如下：

- Step1：对物品进行排序；
- Step2：最优子结构 $OPT[i][j]$ ，表示剩余空间为*j*时第*i*个物品的最优解，从第*i*个物品开始，每个物品可以选*k*个

- Step3：递推关系式 $OPT[i][j]=\max\{OPT[i-1][j-k*wi]+k*vi\}$ s.t. $j-k*wi \geq 0$ and $0 \leq k \leq mi$

上述的三种背包问题都用的动态规划算法，当然卜老师课上的动态规划算法实例很多，也更具有建设性，背包问题算是入门级的问题，也希望对各位有些许帮助。由于本人的算法水平也比较薄弱，若有疏忽或谬误，还望您在评论区不吝赐教，感谢您的时间！

喜欢此内容的人还喜欢

LOA公众号关闭通知

LOA算法学习笔记



杨紫母亲太低调！原来是《甄嬛传》中的她，怪不得女儿这么出色！

一级消防工程师



教育部曝光第八批8起违反教师职业行为十项准则典型案例

在线学术论坛

