

背包问题

蔡天乐

<https://github.com/ctl1111>

背包问题

- 01背包问题
- 完全背包问题
- 多重背包问题
- 混合背包问题
- And so on...

01背包

问题描述：有 N 件物品和一个容量为 V 的背包。放入第 i 件物品耗费的费用是 C_i ，得到的价值是 W_i 。

求解将哪些物品装入背包可使价值总和最大。

01背包

$F[i][v]$ 表示前 i 件物品恰放入一个容量为 v 的背包可以获得的最大价值。

则其状态转移方程便是：

$$F[i][v] = \max(F[i-1][v], F[i-1][v - c_i] + W_i)$$

01背包

问题：

- " i "这一维是否必要？（不必要，可以压缩到只有一维）
- 如何保证每个物品最多放一次？（按照价值从大到小更新状态，以小的状态更新大的状态，从而每个被更新的状态来自于一个尚未更新的状态，也即来自一个表示当前物品没有被放入背包的状态，具体见后面的伪代码）

$F[i-1][v] \longrightarrow F[i][v]$

$F[i-1][v-c_i]+W_i$



01背包

伪代码：

```
F[0..V]=0
for i:=1..N
  for v:=V..C_i
    F[v]=max(F[v], F[v-C_i]+W_i)
```

完全背包

问题描述：有 N 种物品和一个容量为 V 的背包，每种物品都有无限件可用。放入第 i 种物品的费用是 C_i ，价值是 W_i 。

求解：将哪些物品装入背包，可使这些物品的耗费的费用总和不超过背包容量，且价值总和最大。

转化为**01**背包？

完全背包

复杂度问题：

- 01背包： $O(VN)$
- 用01背包做完全背包：

$$O\left(V \sum_{i=1}^N \left\lceil \frac{V}{c_i} \right\rceil\right)$$

复杂度有点高（可以有 $O(VN)$ 的算法）

完全背包

伪代码：

```
F[0..V]=0
for i:=1..N
  for v:=C_i..V
    F[v]=max(F[v], F[v-C_i]+W_i)
```

Problems:

- 01背包中要求恰好装满如何实现？
- 转化为01背包解完全背包问题能否优化？

我有特殊的初始化技巧（对应问题一）

$$F[0] = 0, F[1..V] = -\infty$$

那么仅有从 $F[0]$ 更新而来的 $F[V]$ 不是 $-\infty$ ，从而满足了恰装满的要求。

我还有特殊的分治技巧（对应问题二）

把第 i 种物品拆成费用为 $C_i 2^k$ 、价值为 $W_i 2^k$ 的若干件物品，其中 k 取遍满足 $C_i 2^k \leq V$ 的非负整数。

注意这样的拆分使得费用不大于 V 的第 i 种物品的任何取法都能被表示成若干新物品的和。

Remarks：类似的想法可以用来优化很多具有“重复的结构”的问题。

Reference:

- 《背包九讲》

Acknowledge:

- Powered by [Marp](#)