

背包问题与信息压缩

Qizy

2018 年 2 月 2 日

成都石室中学

yongzhengqi@gmail.com

01 背包问题

完全背包问题

多重背包问题

二维费用的背包问题

分组的背包问题

有依赖的背包问题

参考资料

01 背包问题

01 背包问题

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 背包问题

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。

时间复杂度： $O(n \cdot V)$

空间复杂度： $O(V)$

我们只关心容量、费用、价值。而不关心具体放入了哪些物品。

所以我们将容量相等、费用相等、价值相等的方案合并在了一起，只记录出现了多少次，而不记录放入了哪些物品

同时我们通过奇特的转移顺序，把“是否使用过当前物品”这个信息也给省略掉了

完全背包问题

完全背包问题

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。

完全背包问题

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。

时间复杂度： $O(n \cdot V)$

空间复杂度： $O(V)$

与 01 背包一样，我们只关心容量、费用、价值。而不关心具体放入了哪些物品。

所以我们将容量相等、费用相等、价值相等的方案合并在了一起，只记录出现了多少次，而不记录放入了哪些物品。

与 01 背包一样，我们只关心容量、费用、价值。而不关心具体放入了哪些物品。

所以我们将容量相等、费用相等、价值相等的方案合并在了一起，只记录出现了多少次，而不记录放入了哪些物品。

更进一步，01 背包通过奇特的转移顺序来去重，但完全背包没有考虑去重的问题。

实际上我们把“当前方案是否已经使用过当前物品”也给省略掉了

多重背包问题

多重背包问题

有 N 种物品和一个容量为 V 的背包。第 i 种物品最多有 M_i 件可用，每件耗费的空间是 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空间总和不超过背包容量，且价值总和最大。

多重背包问题

有 N 种物品和一个容量为 V 的背包。第 i 种物品最多有 M_i 件可用，每件耗费的空间是 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空间总和不超过背包容量，且价值总和最大。

使用二进制分组的方式，将其转化为 01 背包问题

时间复杂度： $O(V \cdot \sum \log M_i)$

空间复杂度： $O(V)$

更优的判定性算法

层间复制

层内做完全背包

时间复杂度: $O(n \cdot V)$

空间复杂度: $O(V)$

考虑时间复杂度里面多出来的那个 \log

实际上我们是用时间的代价，把“当前物品用了多少件”的信息给省略掉了

考虑时间复杂度里面多出来的那个 \log

实际上我们是用时间的代价，把“当前物品用了多少件”的信息给省略掉了

考虑判定性算法

实际上是把原始问题的价值那一维，腾给了当前物品的剩余件数

二维费用的背包问题

二维费用的背包问题

二维费用的背包问题是指：对于每件物品，具有两种不同的费用，选择这件物品必须同时付出这两种费用。对于每种费用都有一个可付出的最大值（背包容量）。问怎样选择物品可以得到最大的价值。

设第 i 件物品所需的两种费用分别为 C_i 和 D_i 。两种费用可付出的最大值（也即两种背包容量）分别为 V 和 U 。物品的价值为 W_i 。

二维费用的背包问题

二维费用的背包问题是指：对于每件物品，具有两种不同的费用，选择这件物品必须同时付出这两种费用。对于每种费用都有一个可付出的最大值（背包容量）。问怎样选择物品可以得到最大的价值。

设第 i 件物品所需的两种费用分别为 C_i 和 D_i 。两种费用可付出的最大值（也即两种背包容量）分别为 V 和 U 。物品的价值为 W_i 。

DP 数组加一维

时间复杂度： $O(n \cdot V \cdot U)$

空间复杂度： $O(V \cdot U)$

我们必须记录多出来的那个费用
即将所有方案分类的时候得分得更细
于是 DP 数组只能加一维

分组的背包问题

分组的背包问题

有 N 件物品和一个容量为 V 的背包。第 i 件物品的费用是 C_i ，价值是 W_i 。这些物品被划分为 K 组，每组中的物品互相冲突，最多选一件。求解将哪些物品装入背包可使这些物品的费用总和不超过背包容量，且价值总和最大。

分组的背包问题

有 N 件物品和一个容量为 V 的背包。第 i 件物品的费用是 C_i ，价值是 W_i 。这些物品被划分为 K 组，每组中的物品互相冲突，最多选一件。求解将哪些物品装入背包可使这些物品的费用总和不超过背包容量，且价值总和最大。

使用奇特的转移顺序

时间复杂度： $O(n \cdot V)$

空间复杂度： $O(V)$

相较于暴力，我们使用奇特的转移方式，缩掉了“是否已经选择过第 i 组的物品”这个信息

有依赖的背包问题

有依赖的背包问题

这种背包问题的物品间存在某种“依赖”的关系。也就是说，物品 i 依赖于物品 j ，表示若选物品 i ，则必须选物品 j 。为了简化起见，我们先设没有某个物品既依赖于别的物品，又被别的物品所依赖；另外，没有某件物品同时依赖多件物品。

有依赖的背包问题

这种背包问题的物品间存在某种“依赖”的关系。也就是说，物品 i 依赖于物品 j ，表示若选物品 i ，则必须选物品 j 。为了简化起见，我们先设没有某个物品既依赖于别的物品，又被别的物品所依赖；另外，没有某件物品同时依赖多件物品。

树内做一次 01 背包，树间再做一次分组背包

时间复杂度： $O(n \cdot V^2)$

空间复杂度： $O(V)$

主件的附件仍然可以具有自己的附件集合。限制只是每个物品最多只依赖于一个物品（只有一个主件）且不出现循环依赖。

较一般的问题

主件的附件仍然可以具有自己的附件集合。限制只是每个物品最多只依赖于一个物品（只有一个主件）且不出现循环依赖。

树内树形 DP 一次，树间再做一次分组背包

时间复杂度： $O(n \cdot V^2)$

空间复杂度： $O(V)$

考虑树间做的那一次分组背包：我们把树内的“具体选择方案”给压缩掉了，只留下了“花费”和“收益”

参考资料

[1] 背包问题九讲