

浅谈几类背包题

徐持衡

浙江省温州中学

引言

- 背包问题作为一个经典问题在动态规划中是很基础的一个部分，然而以 0-1 背包问题为原题，衍生转变出的各类题目，可以说是千变万化，当然解法也各有不同，如此就有了继续探究的价值。

我的论文

- 引言
- 背包的基本变换
 - ① 完全背包
 - ② 多次背包
 - ③ 单调队列优化
- 其他几类背包问题
 - ① 树形依赖背包（选课）
 - ② PKU3093
- 总结

多次背包

- **多次背包问题：**给定 n 种物品和一个背包。第 i 种物品的价值是 W_i ，其体积为 V_i ，数量是 K_i 件，背包的容量为 C 。可以任意选择装入背包中的物品，求装入背包中物品的最大总价值。

单调队列

- 对于一个左右边界都只增不降的区间最值，可以用单调队列来做到总效率 $O(n)$ 的维护。
- 如果要用单调队列来优化多次背包，就必须在多次背包问题中挖掘出一个维护区间最值的子问题。

单调队列优化

- 对于多次背包，用最常见的状态表示是用 $F[i, j]$ 表示前 i 种物品，总体积不超过 j 的最大价值总和。
- 当前只考虑第 i 种物品，假设体积 v ，价值 w ，数量 k 。
- 由于对于体积 j ，与其相关的只有那些对 v 的余数与 j 相同的体积，所以再按照体积 j 对 v 的余数分为 v 份。

单调队列优化

- 我们可以把每一份分开处理，假设现在要考虑余数为 d 的部分。

- 用 j 来标号，规定编号 j 所对应的体积是

编号 j	0	1	2	3	4
对应体积	d	$d+v$	$d+2*v$	$d+3*v$	$d+4*v$

- 显然编号 j 可以从编号 $j-k$ 到 j 中的任意一个转移而来，因为相邻的体积正好相差 v 。

单调队列优化

- 区间 $[j-k, j]$ ，这个区间是随着 j 的递增而左右边界都递增的区间。
- 但是注意到由于不同编号对应的体积也是不一样的，显然体积大的价值也会大于等于体积小的，直接比较是没有意义的，所以还需要做一定的修正。

单调队列优化

- 比如可以把体积 $d+j^*v$ 都退化到 d ，也就是说用 $F[i-1, j^*v+d] - j^*w$ 来代替原来的价值进行比较大小。
- 这样就可以用单调队列来优化了，对于每件物品的转移均摊 $O(C)$ ，所以得到 $O(n^*C)$ 的算法。

树形依赖背包（选课）

- 树形依赖背包问题：给定 n 件物品和一个背包。第 i 件物品的价值是 w_i ，其体积为 V_i ，但是依赖于第 X_i 件物品（必须选取 X_i 后才能取 i ，如果无依赖则 $X_i=0$ ），依赖关系形成森林，背包的容量为 C 。可以任意选择装入背包中的物品，求装入背包中物品的最大总价值。

泛化物品

- 这个概念最初是由 DDenigi 在《背包九讲》中提出。
- **定义：考虑这样一种物品，它并没有固定的费用（体积）和价值，而是它的价值随着你分配给它的费用（体积）变化而变化。**

泛化物品

- 泛化物品可以用一个一维数组来表示体积与价值的关系 $G[j]$ 表示当体积为 j 的时候，相对应的价值为 $G[j]$ ($C \geq j \geq 0$)。
- 显然，之前的背包动规数组 F_i ，就是一件泛化物品，因为 $F_i[j]$ 表示的正是体积为 j 的时候的最大价值。同样的，多件物品也是可以合并成一件泛化物品。

泛化物品

- 泛化物品的和：
- 把两个泛化物品合并成一个泛化物品的运算，就是枚举体积分配给两个泛化物品，满足：
：
- $G[j] = \max \{ G_1[j-k] + G_2[k] \}$
($C \geq j \geq k \geq 0$)
- 把两个泛化物品合并的时间复杂度是 $O(C^2)$ 。
- 这个概念也引自《背包九讲》。

树形依赖背包（选课）

- 对于这个问题，我们可以把每棵子树看作是一个泛化物品，那么一棵子树的泛化物品就是子树根节点的这件物品的泛化物品与由根所连的所有子树的泛化物品的和。
- 这就是一个 $O(n \cdot C^2)$ 的算法。

泛化物品

- 泛化物品与一件物品的和：
- 把一个泛化物品与一件物品合并成一个泛化物品，可以用类似于 0 - 1 背包经典动规的方法求出。
- 泛化物品的并：
- 因为两个泛化物品之间存在交集，所以不能同时两者都取，那么我们就需要求泛化物品的并，对同一体积，我们需要选取两者中价值较大的一者，效率 $O(C)$ 。

树形依赖背包（选课）

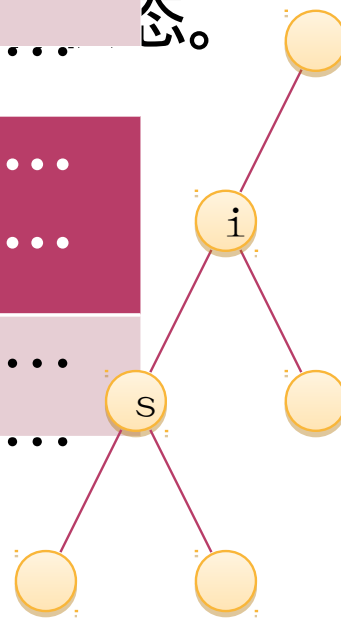
- 考虑对以 i 为根的子树的处理，假设当前需要处理 i 的一个子节点 s ，体积 v ，价值 w 。
- 如果我们在当前的 F_i 中强制放入物品 s 后作为以 s 为根的子树的初始状态的话，那么处理完以 s 为根的子树以后， F_s 就是与 F_i 有交集的泛化物品（实际上是 F_s 包含 F_i ），同时， F_s 必须满足放了物品 s ，即 $F_s[j] \ (v > j \geq 0)$ 已经无意义了，而 $F_s[j] \ (C \geq j \geq v)$ 必然包含物品 s 。
- 下一步只要求 F_s 与 F_i 的并，就完成了对一个个子节点的处理。

树形依赖背包（选课）

- 对于当前节点 i ，我们找到一个子节点 s 。

编号	0	1	2	3	4	5	...
F_i	a_0	a_1	a_2	a_3	a_4	a_5	...

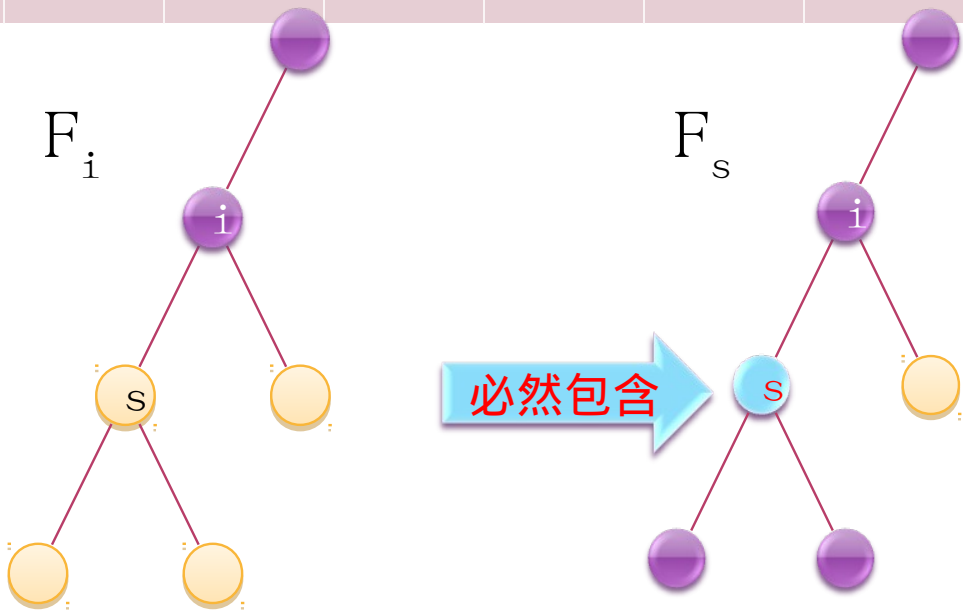
编号	0	1	2	3	4	5	...
F_s	a_0	a_1	a_2	a_3	a_4	a_5	...



树形依赖背包（选课）

- 递归后就得到了最终的 F_s 。

编号	0	1	2	3	4	5	...
F_s	b_0	b_1	b_2	b_3	b_4	b_5	...



树形依赖背包（选课）

- 再通过求 F_i 与 F_s 的并来完成对子节点 s 的更新。

编号	0	1	2	3	4	5
F_s	b_0+w	b_1+w	b_2+w	b_3+w	b_4+w	b_5+w



编号	v	$v+1$	$v+2$	$v+3$	$v+4$	$v+5$
F_i	a_v	a_{v+1}	a_{v+2}	a_{v+3}	a_{v+4}	a_{v+5}

- 最后得到新的 F_i ，就完成了对子节点 s 的更新。

树形依赖背包（选课）

```
1  PROCEDURE DEAL i , C
2  FOR s: = 1 TO n
3  IF s 是 i 的子节点 THEN
4     $F_i \rightarrow F_s$ 
5    DEAL s ,  $C - V_s$       // 背包容量减小  $V_s$ 
6    FOR K: =  $V_s$  TO C      // 求两者的并
7       $\text{Max} ( F_i[ k ] , F_s[ k - V_s ] + W_s ) \rightarrow F_i[ k ]$ 
8    END FOR
9  END IF
10 END FOR
11 END
```

树形依赖背包（选课）

- 对于每一个节点都需要更新一次，所以总效率是 $O(n * C + n^2)$ 。
- 其中找子节点的可以用记边的方式来实现，总效率 $O(n * C)$ 。

树形依赖背包（选课）

- 用这个算法，可以把《选课》这一类 treedp 题优化到 $O(n \cdot C)$ ，亦可以作为 noip06 《金明的预算方案》的 $O(n \cdot C)$ treedp 解法。

总结

- 由背包转变出的题有很多，其中不乏难题，今天给大家介绍了两种最常见的变形，最后都以 $O(n \cdot C)$ 的算法解决了。
- 就目前来说，背包类的题目还有很多没有得到很好的解决，等待着大家去继续探索研究。

- 特别感谢
- 白彦博
- 金圣玺
- 冯一

- 参考资料
- Ddengi 的《背包九讲》

谢谢观看

徐持衡

浙江省温州中学