多角度思考 创造性思维

---- 运用树型动态规划解题的思路和方法探析

江苏省南京外国语学校　陈瑜希

引入

- 信息学竞赛中通常会出现这样的问题：给一棵树，要求以最少的代价（或取得最大收益）完成给定的操作
- 有很多问题都是在树和最优性的基础上进行了扩充和加强，从而变成了棘手的问题
- 这类问题通常规模较大，枚举算法的效率无法胜任，贪心算法不能得到最优解，因此要用动态规划解决

引入

- 在近几年信息学竞赛中，需要运用树型动态规划解决的问题频繁出现
- 这些问题变化繁多，各类思想精华渗透其中，对选手分析问题的能力和解题创造性思维有着较高的要求，因此它在竞赛中占据了重要地位

引入

- 在此，我将分析近几年国际比赛、全国比赛中的树型动态规划问题，重点探讨几种树型动态规划问题的解法，并从这些问题的分析过程中，提炼出解决这类问题的思想方法——多角度思考，创造性思维。

- 旨在论述解决问题的思维过程，而不仅仅是解题方法

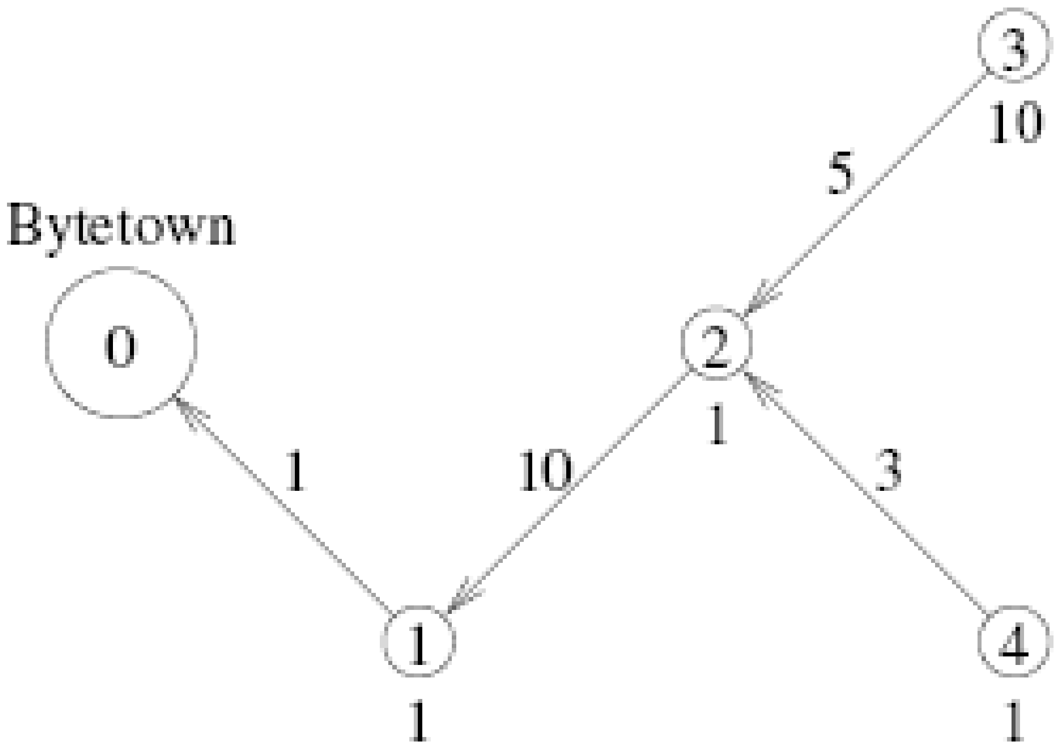
例题解析

- NOI03 逃学的小孩
- IOI05 河流
- NOI06 网络收费
- POI04 山洞

问题描述

- n 个伐木的村庄
- 在 0 号结点有一个巨大的伐木场，木料被砍下后，顺着河流而被运到 0 号结点的伐木场
- 为了减少运输木料的费用，再额外建造 k 个伐木场
- 这些伐木场建造后，木料可以在运输过程中第一个碰到的新伐木场被处理。

问题描述



。

问题抽象

- 本题的题意很明确，即建立 k 个伐木厂，使得把所有木材运送到最近的祖先伐木厂的费用最小。
- 由于题目给定的是一棵树，数据规模又比较大，很容易联想到树型动态规划。

状态的确立

- 首先必须有的是当前点及以当前点为根的子树中，一共建立了多少伐木厂，但是这显然是不够的，因为这个状态中没有任何与伐木厂位置相关的信息。因此我们还需要再加一维。加上有关伐木厂的位置的信息。

$f(from, to\_, kleft)$

表示把以 from 为根的子树中建立 kleft 个伐木厂，把木材全部运送到最近的祖先伐木厂，所需要的费用，并且 from 有一个祖先伐木厂为 to_ 。

（注意到这里 to_ 仅仅是 from 的祖先伐木厂，而未必是 from 的最近祖先伐木厂，这是为什么呢？）

状态的转移

- 状态转移分 2 种情况讨论:
-    在 from 建立伐木厂
-    不在 from 建立伐木厂

状态的转移

- 在 from 建立伐木厂：
- 即分配 kleft 个伐木厂给 from 的子结点，使得费用最小。这分配的过程，也就相当于背包问题。

- h1 是用来解背包问题的临时数组，son 是 from 的儿子结点。

$$h1(i,k) = h1(i-1, k-j) + f(son, from, j)$$

状态的转移

- 不在 from 建立伐木厂：
- 依然是分配 kleft 个伐木厂给 from 的子结点，使得费用最小。
- 不过不同的是，权不是 $f(son, from, j)$ 而是 $f(son,to\_,j)$，因为 from 处不一定有伐木厂。
- h2 是用来解背包问题的临时数组，son 是 from 的儿子结点。

$$h2(i,k)=h2(i-1,k-j)+f(son,to\_,j)$$

状态的转移

- 最后

$$f(from, to\_, kleft)$$
$$= \min\{h1(son\_num, kleft), h2(son\_num, kleft)\}$$

效率分析

- 由于状态是 3 维的，而转移时需要枚举 k 、 son 、和 j ，看上去时间复杂度巨大。
- 这是为什么？
- 刚才的分析通过状态量和转移量相乘来分析效率，每一维都不到 N 。
- j 的总数是 n ， son 的总数也是 n ，所以虽然要枚举 3 个，但是总的运算量是一定的，时间复杂度为 $O(n^3k)$。

回顾

- 本题的动态规划是多维的，要通过分析建立状态
- 在兄弟结点之间 题的思想进行第二次
- 不是单纯的根据状态量和转移量分析时间复杂度，而是根据转移总量分析

总量分析
均摊分析

例题解析

- NOI03 逃学的小孩
- IOI05 河流
- NOI06 网络收费
- POI04 山洞

问题描述

- $M=2^N$ 个点，构成一个满二叉树
- 配对收费：对于每两个用户 $i, j\ (1 \le i < j \le 2^N)$ 进行收费。
- 用户可以自行选择两种付费方式 A 、 B 中的一种，收取的费用等于每两位不同用户配对产生费用之和。

问题描述

I 付费方式	*J* 付费方式	nA 与 nB 大小关系	付费系数 *k*	实际付费
A	A		2	
A	B		1	
B	A	$nA<nB$	1	
B	B		0	
A	A		0	$k * F_{i,j}$
A	B		1	
B	A	$nA \geq nB$	1	
B	B		2	

问题描述

- 对于用户 i，如果他/她想改变付费方式（由 A 改为 B 或由 B 改为 A），就必须支付 C_i 元给网络公司以修改档案。
- 给定每个用户注册时所选择的付费方式以及 C_i，试求这些用户应该如何选择自己的付费方式以使得总费用最少（更改付费方式费用 + 配对收费的费用）。
- $N \le 10$

问题转化

单点收费

- 配对收费的规则
  - B 较多时， AA 收费系数为 2
  - AB 收费系数为 1
  - BB 收费系数为 0
  - 其他情况反之
- 设想：
  - B 较多时，在每一个 A 结点上有 1 个收费系数
  - 否则在每一个 B 结点上有 1 个收费系数

状态的确立

- 状态的设计应该与以 i 点为根的子树中 A 的个数有关，但仅仅这样是不够的，因为这些点是按 A 收费还是 B 收费还与以它每个祖先为根的子树中， A 多还是 B 多有关。因此，这也是需要记录的。

$$f(i,j,k)$$

点 i 所管辖的所有用户中，有 j 个用户为 A ，在 i 的每个祖先 u 上，如果 na<nb ，则标 0 ，否则标 1 ，这个数列可以用二进制表示，用 k 记录，在这种情况下的最少花费。

状态的转移

- 状态转移方程，其实就是将 j 个用户分配给 i 的左右子节点，并更改 k

$$f(i, j, k) =$$

$$\min\{ f(left, u, k+s) + w1, f(right, j-u, k+s) + w2\}$$

状态的转移

- 当 i 是叶节点时，可以根据 k 算出 i 与其它用户配对收费时所要交的费用，再根据 i 的初始情况加上 AB 的转化费用。

效率分析

- 粗略看来，空间复杂度是$O(m^3)$，时间复杂度是$O(m^4)$，对于本题的数据可以同时超空间和超时间了。
- 不过这只是很粗略的分析，这些状态中有很多是取不到的。

效率分析

- 把根节点看做第 0 层，叶节点就是第 n 层，对于任意的第 i 层，A 用户的个数最多只有 2^{n-i} ，而祖先结点只有 i 个，即 k 最大只有 2^i ，把 $f(i,j,k)$ 的后两维合并成一维，空间复杂度其实只有 $O(m^2)$ 。

效率分析

- 再看时间复杂度，对于第 i 层，有 2^i 个结点，每个节点的状态数是 O(m) 的，状态转移的复杂度是 $O(2^{n-i})$，所以每层的复杂度是 $O(m^2)$。
- 总共有 n 层，所以状态转移的时间复杂度是 $O(m^2 n)$。

回顾

- 对收费规则进行转化
- 对时间复杂度进行均摊分析

- 第二点在树型动态规划中有着广泛的运用，本题通过粗略的分析会超时，但是仔细分析之后，发现时间复杂度比粗略的分析少了一维

总结

- 这 **2** 个例题从不同方面阐述了树型动态规划的解题方法，如：
  - 多维动态规划
  - 兄弟结点之间通过类似背包的思想进行第二次动态规划
  - 对复杂度的均摊分析
- 这些方法在比赛中有着广泛的运用

总结

- 在此过程中，我认识到：面对陌生的问题，一方面要深入分析问题的属性，挖掘问题的本质，另一方面要多角度思考，抓住问题的特殊性，从而创造出正确的解题思路。
- 不过，这类问题变化繁多，我只是提供了一些解题的方法和思路，抛砖引玉，重要的是平时的不断积累和总结