构造类问题的若干解题方法

范斯喆

浙江省诸暨市海亮高级中学

2025 年 5 月 16 日

前言

在近年的算法竞赛中，构造题的出现越来越频繁……对不起拿错剧本了。
这场讲课主要是想给大家分享一些构造题中比较常用的想法或思路。

调整法

邓老师 2021 年的集训队论文中介绍过一种调整法的思路：先给出一个满足部分条件的方案，再不断调整方案，使其变得更满足条件。

实际上，构造题中还有两种调整的思路。

第一种调整

先构造一个方案，然后发现方案存在问题，通过调整改进之。

[IOI 2020] 网络站点

这是一道通信题。
给定一棵 n 个点的树，以及 k。你需要给每个点一个编号，编号必须是 $0 \sim k$ 中的数，且不能相同。
然后会有 q 次调用，每次给定 s, t，表示起点和终点的编号。然后给定起点的所有邻居的编号，让你找出那个离终点最近的邻居。
$2 \leq n \leq 1000$，$k \geq n - 1$。

题解

我们先考虑直接用 dfs 序。dfs 序可以区分各个儿子的子树，但是区分不了“最后一个儿子的子树”和“子树外”。
考虑如何改进。如果我们知道当前节点的出栈序，就可以区分这两种情况了。
所以奇数层用入栈序，偶数层用出栈序，就做完了。

CF1930H Interactive Mex Tree

这是一道交互题。
给定一棵 n 个点的树，q 次询问。
所有询问开始前，你需要构造两个 $1 \sim n$ 的排列 p_1, p_2。
每次询问时，会生成一个 $0 \sim n-1$ 的排列 A，作为点权（不会告诉你）。然后要询问一条路径上的 mex。
为了回答问题，你可以查询至多 5 次。每次查询，你需要给出 x, l, r，交互库会返回 $\min_{i \in [l,r]} A_{p_{x,i}}$。
$2 \le n \le 10^5$，$1 \le q \le 10^4$，$n \times q \le 3 \times 10^6$。

因为点权是排列，所以查询路径 mex 等价于查询路径之外的点权 $\min$。
设路径为 (x, y)，它们的 LCA 为 z。
同样，我们考虑 dfs 序，但是处理不了 z 到 x 或者 z 到 y 的部分。
如果第二个排列是出栈序，就能处理了。

第二种调整

有时候题目会限制某个量“恰好为 k”。这时可以尝试求出这个量的上下界，并证明上下界之间的数都能取到。然后从取到上界或下界的方案开始，逐步调整到给定值。

更抽象地说，这种思路实际上是估计“最好情况”和“最坏情况”，依此来推出所有“中间情况”。所以“恰好为 k”这样的条件并不是必要的。

给定 n, d，要构造一棵 n 个点的二叉树，满足所有点到根的距离之和为 d。
可能无解。
$2 \leq n, d \leq 5000$。可以做到线性。

题解

距离之和最小的是完全二叉树。
距离之和最大的是链。
考虑从完全二叉树向链调整。每次选出一个不在链上的叶子 x。
设链底的深度为 d_1，这个叶子的深度为 d_2，则把 x 接到链底会让总深度增加 $d_1 - d_2 + 1$。
如果增加之后不超过 d，就可以这样做；否则可以算出把它接到链上的哪个位置。
复杂度 $O(n)$。

Puzzle: Nurikabe

有一个 $n \times m$ 的矩形网格。给定 x, y, z，你需要将每个格子涂为黑色或者白色，使得：

1. 所有黑色格子四联通；
2. 所有白色格子四联通；
3. 不存在 2×2 的子矩形，使得这个子矩形中的格子全是黑色；
4. (x, y) 这个格子是白色；
5. 白色连通块的大小为 z。

构造方案。

$n \times m \leq 10^6$。

题解

考虑对于 (x, y)，找到有解的 z 的最小值和最大值。最大值显然是 $n \times m$。
如果 $n = 1$，最小值为 $\max(y, m - y + 1)$。
如果 $n = 2$，最小值为 $m - 2 + [y = 1 \vee y = m]$。
据此，我们只需要做 $n, m \geq 3$ 的情况。
先考虑求出下界。注意到 z 个白色格子最多覆盖 $2z + 2$ 个内部点，因此有 $2z + 2 \geq (n - 1)(m - 1)$。
观察发现，只要 (x, y) 不在边界上，这个下界总是可达的。在边界上会 $+1$，在角落上会 $+2$。
从下界方案开始调整，只需固定一个删除黑格的顺序即可。复杂度 $O(nm)$。

有一个 $n \times n$ 的无向网格图。
你需要构造 $2n$ 条路径:

- 第 i 条路径的起点为 $(1, i)$，终点为 (n, p_i);
- 第 $n+i$ 条路径的起点为 $(i, 1)$，终点为 (q_i, n);

其中 p 和 q 都是长度为 n 的排列。
定义一个构造方案的代价为：每条边被路径覆盖次数的最大值。
求一个代价最小的方案。
$1 \leq n \leq 200$。

题解

考虑最好情况，也就是 $p_i = q_i = i$ 的情况。发现答案为 1，且只有这种情况答案为 1。
考虑对应的最坏情况，也就是 $p_i = q_i = n - i + 1$。发现可以构造出代价为 2 的方案。
具体地，对于从 $(1, i)$ 到 $(n, n - i + 1)$ 的路径，我们先从 $(1, i)$ 向下走到 (i, i)，再横向移动到 $(i, n - i + 1)$，然后继续向下到 $(n, n - i + 1)$。
$(i, 1)$ 到 $(n - i + 1, n)$ 的情况类似。易证每条边都会被经过两次。
从最坏情况开始，逐步调整到最初的 p, q。注意到如果 $p_i > p_{i+1}$，则 $(1, i) \to (n, p_i)$ 和 $(1, i + 1) \to (n, p_{i+1})$ 一定有交。类似 LGV 引理，在交点处交换路径，就相当于交换了 p_i 和 p_{i+1}。
复杂度 $O(n^3)$。

增量法/归约法

这种方法比较类似数学中的数学归纳法。
增量法的思想是，假设已经有了一个构造方法，将它扩展到更大的情况。
归约法的思想是，找到构造目标的某个部分，将它先构造掉，然后就只需考虑删掉它之后的情况。

经典题

给定一个 n 个点的竞赛图，求它的一条哈密顿路径。
$1 \le n \le 5 \times 10^3$。

题解

考虑增量法。假设已经对点 $1 \sim n-1$ 的导出子图完成构造，现在要加入点 n。
不妨设已经构造出的哈密顿路径为 $p_1, p_2, \cdots, p_{n-1}$。
如果存在 $n \to p_1$ 的边，那就直接把 n 接在前面。
如果存在 $p_{n-1} \to n$ 的边，那就把 n 接在最后。
否则，一定存在某个 i，使得存在 $p_i \to n$ 和 $n \to p_{i+1}$ 的边。把 n 塞到他俩中间即可。

[CCO 2020] Travelling Salesperson

给定一个 n 个点的完全图，每条边都是红色或者蓝色。
对于每个点，你需要构造一条从该点出发，经过每个点至少一次，且长度最短的路径。要求这条路径只能“切换”一次颜色。
$1 \leq n \leq 2 \times 10^3$。

题解

不妨猜测可以构造哈密顿路径。
考虑增量法，从 $n-1$ 推到 n。
设红蓝交界点为 m。如果 n 和 m 的连边是红色，那么 n 可以接到 m 的蓝色那一边。否则就接到红色那一边。注意插入后可能要改变交界点的位置。
“从某个点出发”的限制是容易满足的。复杂度 $O(n^2)$。

CF1019C Sergey's problem

给定一张 n 个点 m 条边有向图，可能有重边，但没有自环。
你需要选出若干个点，使得其中的点两两之间没有连边，且任意没被选的点都能通过被选的点走至多两步到达。
$1 \leq n, m \leq 10^6$。

题解

直接归约是不行的，因为不能单独决策某个点是否被选。
考虑增量法。加入一个点时，如果它有入点选了，就不用管；否则，如果出点都没选，它就得选，如果某个出点选了就会出问题。
于是倒过来归约。每次删掉某个点和它的所有出点，做子问题。
做完之后加入这个点即可，这样所有出点一定都没被选。
复杂度 $O(n)$。

[COTS 2022] Kraljice

有一个 $n \times n$ 的棋盘。你要依次在棋盘上放置若干个皇后，使得每个皇后被放置时它都会被偶数个皇后攻击。
你想最大化放置皇后的数量，并构造一组方案。
$1 \le n \le 2^{10}$。

题解

在矩阵上的问题中，我们一般有三种思路：按照行/列考虑，按照对角线考虑，或者从内向外/从外向内考虑。
注意到，如果我们填满棋盘最外围一圈，那么中间每个格子都会被偶数个皇后攻击。也就是说这一圈不会对中间造成影响。
于是可以考虑从 $n \times n$ 的棋盘归约到中间 $(n-2) \times (n-2)$ 的子棋盘。
经过手玩发现，如下填数方式是可行的：

1	3	4	7	15
9			14	8
10				11
13				12
17	2	5	6	16

实际上，每归约一层都要左右翻转一下。$n \leq 4$ 时无法构造，退出递归即可。

基础构造练习题 1

有一列实数，对于每一次操作，可以选择两个实数，把它们同时变为两数之积。

给定数列的长度 n，你需要找到一种操作方案，使得对于任意长度为 n 的实数列，按照该方案操作后，数列的每一项都相同。

$2 \leq n \leq 2^{10}$。操作次数越少越好。

题解

首先，如果 n 是奇数一定无解。因为最终的数一定是把 n 个数分成 $\frac{n}{2}$ 对，每对之间乘起来。
$n=2$ 和 $n=4$ 是好做的。从 n 推到 $2n$ 也是容易的。考虑从 n 推到 $n+2$。
先操作前 n 个数使其相同，然后对后两个数也操作一次。现在序列形如 $x, x, \cdots, x, y, y$。然后按照如下方式操作：

$$
\begin{aligned}
& x, x, x, x, \cdots, x, x, x, y, y \\
\rightarrow\ & x, x, x, x, \cdots, x, x, xy, xy, y \\
\rightarrow\ & x, x, x, x, \cdots, x, x^2y, x^2y, xy, y \\
\rightarrow\ & x, x, x, x, \cdots, x^3y, x^3y, x^2y, xy, y \\
\rightarrow\ & x, x, x^{n-4}y, x^{n-4}y, \cdots, x^4y, x^3y, x^2y, xy, y \\
\rightarrow\ & x, x, x^{n-4}y^2, x^{n-4}y, \cdots, x^4y, x^3y, x^2y, xy, x^{n-4}y^2
\end{aligned}
$$

这样就可以做到 $O(n \log n)$。
事实上，我们可以推广出从 n 推到 $2(k+1)n$ 的方法：把序列拆成 n 个长度为 $2k+2$ 的序列，每个序列做一次，总共 $O(kn)$ 次。
然后对增量方式进行 DP，可以做到 1.1×10^4 次左右。
实际上这个题可以更优，不过主题和增量法关系不大，略。

结语

谢谢大家！