NOI 2018 day1/day2 试题解析

钟子谦

(清华大学 100084)

**NOI 2018 day1/day2 problem analysis**

ZHONG Ziqian

(Tsinghua University, Beijing 100084)

Day1

1 归程

【分析】

首先使用Dijkstra算法求出1号点到每个点x的距离d[x]，那么我们对于每个询问就是要从v开始，经过一些海拔>=p的边走到某个点t，将d[t]更新答案。即我们要询问的是只加入>=p的边时v所在联通块d的最小值。如果不强制在线的话我们只需要将边从大到小加入，使用并查集维护即可。强制在线的话，可以直接将并查集用可持久化线段树等数据结构可持久化，这样复杂度是O(nlog^2n)或O(nlognα(n)) （假设n、m、q同阶，下同）。更简单的做法是建立kruskal重构树，模拟求最大生成树的kruskal过程，每次合并两个联通块时建立一个新的父亲点，点权为当前边的边权。那么我们只需要每次找到v在这个重构树上最高的点权>=p的点，这个子树内的叶子即为可达点。使用树形dp和倍增维护即可，这样就是O(nlogn)的。

2 冒泡排序

【分析】

容易观察出条件等价于最长下降子序列长度不超过2。这是因为如果有一个数左边有比它大的数，右边有比它小的数，这个数必然会被向左交换一次，又向右交换一次，这两步对于这个数的位置影响抵消了，所以浪费了移动次数，一定不合法。类似地可以证明若最长下降子序列长度不超过2一定合法。

如果没有字典序的限制，考虑进行dp，我们记f[i][j]表示前i个数，比当前最大的数小的还没用过的数共有j个的方案数，那么要么选一个比当前最大值更大的数，转移到f[i+1][j+t]（t>=0），否则选取最小的还没用过的数，转移到f[i+1][j-1]（不能选择还没用过的比当前最大值小的其他数，因为后来会与最小的构成一个长度为3的下降子序列）

考虑这个dp的过程，这就相当于在平面上选取了一条从(0,0)到(n,0)的路径，每次横坐标增加一个单位，纵坐标增加一个>=-1的整数单位，需要保证每时每刻纵坐标非负。如果没有纵坐标非负的限制，我们可以简单地用组合数算出答案（相当于n个>=-1的整数和为0的方案数，即n个正整数和为2n的方案数，隔板法），否则的话一定经过了y=-1这一直线，我们将终点关于y=-1对称，即将第一次碰到y=-1后的路径翻转，类似地组合数计算即可。

接下来考虑初始字典序的限制，我们只需要枚举第一次突破限制的位，类似地使用组合数计算即可。预处理出阶乘和逆元，复杂度O(n)。

3 你的名字

【分析】

题意即每次给出一个询问串T，问它有多少个本质不同的子串不是S[l,r]的子串。

对于每个左端点i，显然存在一个界a[i]满足T以i开始的子串中不是S[l,r]子串的恰是结尾>=a[i]的，并且a不降。考虑在S的后缀自动机上匹配T，向后加字符即走ch边，在前端删除字符即为不动或走fail边。在后缀自动机fail树的dfs序上建立可持久化线段树，支持询问某个点right集合中是否有某个区间内的数，这样我们就可以使用two-pointer求出a。接下来建立T的后缀数组，利用height数组扣掉排序后相邻后缀多算的即可。时间复杂度为O(n)或O(nlogn)（假设输入长度为n）。

Day2

1屠龙勇士

【分析】

首先用multiset求出杀死每只龙所用的剑，那么杀死龙的条件即a[i]-x\*atk[i]<=0，且a[i]-x\*atk[i]为p[i]的倍数。不等式限制了x的下界，同余方程可以使用扩展欧几里得算法合并。

2 情报中心

【分析】

题意即选出两条有边相交的链，最大化选出的链的权值和和被至少一条链覆盖的边的边权和之和。

两种情况讨论：两条链的lca相同和两条链的lca不同。

若两条链x1-x2、y1-y2的lca不同，链的交必然是一段祖孙链，如左图。

我们枚举C点，那么x2和y2必须分属C的不同子树（或为C），贡献是两条链的长度和（长度即链上边的边权和）+两条链的权值和-C点深度+max(B点深度，A点深度)

我们对每个点i记录f[i][j]为下端点在i子树内，上端点深度为j的链长度+权值的最大值。使用动态开点线段树维护这个数组，并启发式合并或线段树合并维护即可。实现时可能要维护区间f[i][j]的最大值和f[i][j]+j的最大值。

若两条链x1-x2、y1-y2的lca相同，如左图。

同样地，x1和y1必须分属B的不同子树（或者就为B），贡献为x2和y2的距离+两条链的长度和+两条链的权值和+x1深度\*2+y1深度\*2-B点深度\*2

考虑在每条链的每个端点下挂一个新点，与原来的点连一条长度为链长度+链权值+另一端点深度\*2的边，那么贡献就变为了两条链的新点距离-B点深度\*2。

这就转化成了一个最远点对问题。跨越两个集合的最远点对，端点一定是原来两个集合的直径端点。合并两个集合时新的直径端点也一定是原来两个集合的直径端点。我们用g[i][j]维护i的子树中有一端，lca为j时的新点集合，每次将g数组启发式合并更新答案，并合并集合即可。

时间复杂度O(nlogn)。

3 多边形

【分析】

考虑在子树上进行连通性dp，记录每个子树的根、前k个叶子、后k个叶子的连通性和度数。状态数较少，预处理转移后可以通过。实现十分复杂，并且需要大量的常数优化和和剪枝。

参 考 文 献

[1] 刘汝佳 算法竞赛入门经典（第2版） 北京：清华大学出版社，2014
[2] 王晓东.数据结构：C语言描述.北京：电子工业出版社，2011
[3] 王晓东.计算机算法设计与分析（第4版）.北京：电子工业出版社，2012