ICPC2023 合肥站赛题讲解

周一凡

中国科学技术大学

2023 年 11 月 26 日

预期题目难度顺序

- 签到题：F
- 简单题：E,G,J
- 中档题：B,I
- 较难题：C,D,K,L
- 难题：A,H

F

字符串长度只有 10，所以可以直接读入所有字符串，并进行排序，统计每种字符串出现的数量即可。

E

这里 x 的贡献和 y 的贡献是分离的，因此我们不妨只考虑怎么算 x 的贡献。

对于每种不同的颜色，我们可以把颜色离散化，然后将相同颜色的 x 坐标存在同一个 vector 里，再进行排序。对于某个 vector 的第 i 个数，它的贡献等于：

$$\sum_{j<i}(vec_i - vec_j) + \sum_{j>i}(vec_j - vec_i) \tag{1}$$

这个式子很容易做到在枚举 i 的过程中维护。

二分第 k 长的连续 1 段的长度。假设当前的二分值为 x，那么需要有至少 k 个连续 1 段长度大于等于 x。可以设计 dp 状态 $f_{i,j,0/1}$ 表示当前考虑了序列的前 i 个位置，前面有 j 个长度大于等于 x 的连续 1 段，当前序列的结尾是否也是某个满足条件的 1 段的结尾，所需要的最少修改次数。
有转移：

$$f_{i,j,1} = f_{i-x,j-1,0} + val[i-x+1, i] \quad (2)$$
$$f_{i,j,0} = \min\{f_{i-1,j,0}, f_{i-1,j,1}\} \quad (3)$$

这里 $val[l, r]$ 为把 $[l, r]$ 该段换为全 1 所需的代价，这个统计出该段的 0 或 1 的数量则容易计算出。

首先可以注意到最大的边在原图的最大生成树上。所以我们先求出最大生成树，然后分别从 1 和 n 出发对树进行 dfs，求出 1 或者 n 到其他点的瓶颈边的权值。接着枚举次大的边 (u, v)，再枚举 1 是和 u 连的还是和 v 连的，求出瓶颈边，两者一拼可得结果。
注意特判只通过一条边连接的情况。

B

这个问题相当于要求能将序列拆分成两个不下降的子序列。由 dilworth 定理，这就相当于这个序列的逆序列的最长上升子序列长度不超过 2。
我们现在问题变成了：有多少种不同的序列，它的最长上升子序列长度不超过 2。
可以采用计数 DP 的方法，一开始构造一个空序列，接着将数字按从小到大的顺序插入到序列之中。用 $f_{i,j}$ 表示当前已经把小于等于 i 的数字插入了，其中

B

可以采用计数 DP 的方法，一开始构造一个空序列，接着将数字按从小到大的顺序插入到序列之中。用 $f_{i,j}$ 表示当前已经把小于等于 i 的数字插入了，其中 $LIS = 2$ 的最前面的位置为 j。现在考虑把数字 $i+1$ 插入到序列里。可以发现这些数字肯定不能插入到 j 之后，插到第一个数之前不会改变 LIS 的长度。又由于 $i+1$ 比其他已经插入的数字都大，所以插入到非第一个位置之前的最前面一个 $i+1$ 会变成新的 $LIS = 2$ 的位置。于是枚举：第一个数前面放了 x 个 $i+1$，首个未放到第一个数前面的 $i+1$ 放到了第 k 个数的后面，于是得到递推式：

$$f[i+1][x+k+1] += f[i][j] \times \binom{j-k-1+a[i+1]-x-1}{a[i+1]-x-1} \qquad (4)$$

I

先对 $0 \sim n$ 编号 $a..zA..z$，将每个字母在第一位，第二位出现，以及同时在第一位和第二位出现的次数用哈希值表示。同时求出输入中每个字母第一位，第二位，第一位和第二位同时出现的次数的哈希值。对于哈希值相同的位置，采用搜索的方法，枚举每种排列对应的情况，同时检查这样排列是否能对应上目标情况。

C

把 S 重复一遍得到 S'，一个循环子串要是 S' 的一个子串，就恰在其长度不超过 n 且结束点大于 n 时统计一下。对 S' 建立回文自动机，考虑通过其 fail 树确定每个串的出现次数。每次 extend 时，一个节点贡献 $size$ 当且仅当它对应字符位置大于 n。然后在树上求 $size$，统计答案。

D

先不考虑 $1 \le k \le \frac{l-1}{2}$ 的限制，那么对于一个固定的 k，满足条件的 l 是一个连续的前缀，我们可以二分这个满足条件的前缀。然后再加上第一个限制，对应合法的 l 就是一个区间。将这个区间用差分的方法记录一下。

至于怎么确认一个前缀是否在 k 下是合法的，可以采用字符串哈希的方法，若 $\mathrm{hash}(1, l-2k) + \mathrm{hash}(k+1, l-k) = 2\mathrm{hash}(2k+1, l)$，则 $s_{1,l}$ 是合法的。

时间复杂度 $O(n \log n)$，若常数小可以通过

D

现在我们从小到大枚举 k，维护一个指针表示右侧最长的已经确认平衡的前缀长度，每次枚举一个新的 k，考虑是否延申这个指针。时间复杂度 $O(n)$

K

注意到一个区域的价值只会收到最大值和次大值的影响，其他点不影响区域的价值。于是不难猜想一个区域只需要保留最大值和次大值作为端点的这条路径即可。

现在问题变成了，给出一棵树，把树划分成若干路径，每条路径的价值等于端点之中较小的一个，所对应的最大价值和。

可以考虑 DP：用 $f_{i,j}$ 表示以 i 为根的子树，其中延申到 i 的链另一个端点的价值为 j，其他部分的最大价值和。g_i 表示以 i 为根的子树，能构成的最大价值。

用 $f_{i,j}$ 表示以 i 为根的子树，其中延申到 i 的链另一个端点的价值为 j，其他部分的最大价值和。g_i 表示以 i 为根的子树，能构成的最大价值。转移的时候，依次枚举 i 的每个儿子 son，对 f 有转移

$$f_{i,j} = \max(f_{i,j} + g_{son}, f_{son,j} + \sum_{k \in other\ son} g_k) \tag{5}$$

对 g 有转移

$$g_i = \max(g_i, f_{i,j} + f_{son,k} + min(j,k)) \tag{6}$$

这两个信息都可以用线段树合并维护，时间复杂度 $O(n \log n)$

L

每个注意的人都会有唯一一个使其从未注意变为注意的人。这样我们可以构造一个 aware 的链。

建出图的 dfs 树，对 dfs 树的几类边 (u, v) 是否会 aware 进行一下讨论：

- 横向边：此时 u 的 dfn 大于 v 的 dfn，如果 v 是由横向边从 u 传播的，那么这条横向边一定传播链的末尾。
- 前向边：和横向边的讨论类似，前向百年也是传播链的末尾。
- 返祖边：如果传播链包含返祖边，那么传播链成环，矛盾。

于是一条传播链只会包含若干树边以及一条前向边或者横向边。

于是我们考虑对于每个点 i，建出连向 i 的其他结点对 dfs 树的虚树。通过 dp 求出只通过 dfs 树边的传播链的概率。接着枚举横向边、前向边，树边，把概率传播到 i。

虚数结点个数 $O(m)$，时间复杂度 $O(n+m)$

A

$\lfloor \sqrt[3]{x^3} \rfloor = b$ 可以解出 x 的取值范围 $[l, r]$，$x^2 \equiv a \pmod{n}$ 转化为二次多项式的找零点问题 $(x - l)^2 - a = 0$，注意这里的 x 是会远小于 n 的。构造多个具有相同解的多项式，然后使用 LLL 算法找到他们的小系数线性组合，然后因为系数小，x 也小，所以多项式的值会比 n 小，于是可以忽略取模直接二次方程求根公式解出来 x。

H

打表可以发现 f, g 的段数不会很多，实际上只有三万段，所以我们可以想办法求出每一段。
注意到每一段增大的时候，一定是右边四个向下取整的函数之一发生了变化，所以我们可以记录当前所有已经求出的所有左端点，然后通过这些已经求出的左端点，反推出下一个未求出的左端点。对于右端点，它等于左端点减一。如此反复可以推出所有的段。
对于每个询问，在众多段中二分一下即可。