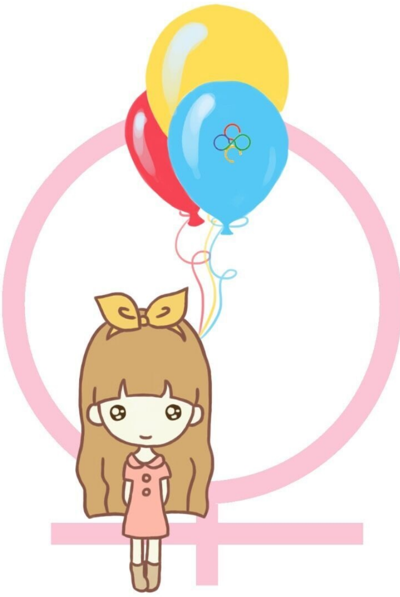
2018 “字节跳动杯” 中国大学生程序设计竞赛女生专场 赛题分析

2018 CCPC-WFINAL



2018 年 5 月 27 日

1 CCPC 直播

Shortest judge solution: 493 Bytes.

按照题意模拟输出结果即可。

2 口算训练

Shortest judge solution: 930 Bytes.

对于一个询问 d，将 d 分解质因数，得到 $d = p_1^{k_1} p_2^{k_2} ... p_m^{k_m}$。若对于每个 p_i，在 $[l, r]$ 里都能找到至少 k_i 个 p_i 的话，则 $a_l \times a_{l+1} \times ... \times a_{r-1} \times a_r$ 是 d 的倍数。

对于 $a_1, a_2, ..., a_n$ 每个数都进行分解质因数，用 v_x 存储质数 x 在序列中从左往右出现的所有位置，那么对于每个询问二分查找即可求出区间里有多少个质数 x。

时间复杂度 $O(n(\sqrt{n} + \log^2 n))$。

3 缺失的数据范围

Shortest judge solution: 718 Bytes.

不难发现随着 n 的增大，$n^a \left(\lceil \log_2 n \rceil\right)^b$ 的值不会减小，反之亦然。二分查找最大的满足 $n^a \left(\lceil \log_2 n \rceil\right)^b$ 不超过 k 的 n 即可。

有两点需要注意：

- 乘法可能溢出 64 位整数，需要特殊处理。
- 求 $\lceil \log_2 n \rceil$ 时不应使用实数计算，这将带来误差，应当使用整数计算。

时间复杂度 $O(\log^2 k)$。

4 寻宝游戏

Shortest judge solution: 1392 Bytes.

假设已经选定了一条路线，那么不考虑具体交换方案，考虑选定一个 $t(t \leq k)$，经过的格子里要有 t 个格子的权值不计入得分，而不经过的格子里要有 t 个格子的权值计入总分。那么每一种交换方案都可以对应这样一个转化。

设 $f_{i,j,x,y}$ 表示从 $(1, 1)$ 出发来到 (i, j)，考虑完前 $i - 1$ 行所有格子以及第 i 行前 j 个格子时，有 x 个经过的格子不计分，y 个不经过的格子计分的情况下，总分的最大值是多少。那么有两种状态转移：

- 往右走一格，直接转移到 $f_{i,j+1,x',y'}$。
- 往下走一格，转移到 $f_{i+1,j,x',y'}$，需要枚举这一行有多少个不经过的格子计分。显然这些格子一定按照权值从大到小贪心选择。

最终答案即为 $\max(f_{n,m,t,t})(0 \leq t \leq k)$，时间复杂度 $O(n^2k^3)$。

5 奢侈的旅行

Shortest judge solution: 1079 Bytes.

考虑一条边 u_i, v_i, a_i, b_i，能经过这条边当且仅当 $\log_2 \frac{level+a_i}{level} \ge b_i$，即 $1 + \frac{a_i}{level} \ge 2^{b_i}$。因此 $level$ 越大越不容易经过这条边。

再考虑最终的总代价，假设依次经过的 k 条边对等级的提升分别为 $a_1, a_2, ..., a_k$，那么代价为 $\log_2(1+a_1) + \log_2 \frac{1+a_1+a_2}{1+a_1} + ... + \log_2 \frac{1+a_1+a_2+...+a_k}{1+a_1+a_2+...+a_{k-1}}$。注意到 $\log_2 a + \log_2 b = \log_2 ab$，所以上式化简后即为 $\log_2(1+a_1+a_2+...+a_k)$，同样也是最终 $level$ 越小越好。

设 f_i 表示从 1 出发到达 i 点的最小可能的等级，用堆优化的 Dijkstra 算法求出所有 f 即可，时间复杂度 $O(m \log n)$。

注意数据比较强，使用复杂度不正确的最短路径算法 (比如 SPFA、加了 SLF 优化的 SPFA、加了 SLF 和 LLL 优化的 SPFA 等) 将会得到 “Time Limit Exceeded”。

6 对称数

Shortest judge solution: 1981 Bytes.

考虑二分答案，问题转化为判定 $[l, r]$ 里是否存在数字出现偶数次。考虑给每种数字随机一个 unsigned long long 范围内的权值 w，则当且仅当该区间内所有数的权值异或和等于 $w_l \oplus w_{l+1} \oplus ... \oplus w_r$ 时，该区间所有数字都出现了奇数次。

用可持久化权值线段树维护每个节点到根路径上每个值域区间内的异或和，每次查询根据 LCA 用 4 棵线段树异或然后在线段树上二分答案即可。

时间复杂度 $O(n \log n)$。

7 赛题分析

Shortest judge solution: 537 Bytes.

按照题意分别求出代码量的最小值然后输出即可。

8 quailty 算法

Shortest judge solution: 793 Bytes.

因为右边每个点恰好连接左边两个点，故可以直接将右边的每个点看成连接左边两个点的一条边，建立一个 n 个点 $\frac{n(n-1)}{2}$ 条边的新图。问题转化为在新图中选择数量最多、权值和最小的一些边，使得这些边存在一组定向方式，使得每个点恰有一条出边。

对于基环树结构，每个连通块点数等于边数，恰好满足以上条件：存在合法定向方式且边数最多。故问题转化为求边权为点权异或值的最小生成基环森林。

这与最小生成树类似，都是拟阵，故可以仿照 Kruskal 算法，按边权从小到大考虑，能选就选。

考虑按二进制从高位到低位分治，显然最高位不同的两个点集内部会优先连边，然后再考虑跨越两个集合的边。那么根据两个点集还需要连几条边可以分析出需要添加多少条跨越的边。

可以证明当两个集合点数都不小于 3 时，再也不能加入任何边，所以如果需要加边，暴力枚举所有边的复杂度为 $O(n)$。

因为一共 $O(\log a)$ 层，每层复杂度为 $O(n)$，所以总时间复杂度为 $O(n \log a)$。

9 SA-IS 后缀数组

Shortest judge solution: 346 Bytes.

设 f_i 表示 suf_i 和 suf_{i+1} 的大小关系，显然因为 suf_{n+1} 为空串，所以 $suf_n > suf_{n+1}$。

对于 $1 \le i < n$，若 $S_i \neq S_{i+1}$，那么 f_i 可以直接通过判断这两个字符的大小得出，否则需要接着比较 suf_{i+1} 和 suf_{i+2}，故此时 $f_i = f_{i+1}$。

从 n 到 1 递推求出所有 f 即可，时间复杂度 $O(n)$。

10 回文树

Shortest judge solution: 1026 Bytes.

因为字符在 $[1, n]$ 随机，故答案期望级别为 $O(n)$，考虑不重不漏地枚举出所有回文路径。

枚举回文中心 (一个点或者一条边)，然后往两侧枚举相同字符扩展即可。注意树的形态并不随机，不能暴力枚举两条边，而需要通过一些技巧加速枚举：比如双指针。

时间复杂度 $O(n \log n)$。

11 代码派对

Shortest judge solution: 812 Bytes.

矩形的交集如果非空，那么仍然是一个矩形。考虑求出经过每个格子 (i, j) 的矩形数量 $f_{i,j}$，f 可以通过二维前缀和在 $O(n + m^2)$ 的时间内求出，则 $ans = \sum C(f_{i,j}, 3)$。

上述方法会重复统计，这些多出来的部分一定是交集覆盖了 $(i, j-1)$ 或者 $(i-1, j)$。减去交集同时覆盖 $(i, j-1)$ 和 (i, j) 的答案，再减去交集同时覆盖 $(i-1, j)$ 和 (i, j) 的答案，那么交集同时覆盖 $(i-1, j-1)$ 和 (i, j) 的答案会被多减一次，再加回来即可。按照该方法，每个方案都将在交集的左上角格子处被恰好统计到一次。

对于计算交集同时覆盖 $(i, j-1)$ 和 (i, j) 的答案，只需要把所有矩形的边长减小 1，其它几项答案的计算同理。

时间复杂度 $O(n + m^2)$。