

The 17th Zhejiang Provincial Collegiate Programming Contest Editorial

Prepared by Zhejiang University

2020 年 10 月 17 日

Overview

	Easiest										Hardest	
Idea	K	A	C	I	F	G	E	H	B	D	L	J
Coding	K	I	B	G	A	C	E	F	L	D	H	J
Summary	K	A	C	I	F	G	E	B	H	D	L	J

K. Killing the Brute-force

Shortest Judge Solution: 330 Bytes

Description

给定标程和暴力在 $n = 1, 2, \dots, m$ 时的运行时间，找到最小的 n 使得三倍标程时限可以卡掉暴力。

- $m \leq 50$ 。

Solution

- 签到题，尝试所有的 n 即可。
- 时间复杂度 $O(m)$ 。

A. AD 2020

Shortest Judge Solution: 957 Bytes

Description

给一个日期的区间，问有多少个日期含有指定子串。

- $2000.01.01 \leq \text{日期} \leq 9999.12.31$ 。
- 测试数据组数 $\leq 10^5$ 。

Solution

- 预处理出 2000.01.01 到每一天的答案。
- 每次查询时利用两个天数对应的答案相减，时间复杂度 $O(1)$ 。

C. Crossword Validation

Shortest Judge Solution: 1050 Bytes

Description

检查一个 $N \times N$ 的 Crossword Puzzle 是否合法，若合法需要统计里面所有极长单词的分数之和。

- $N \leq 1000$ 。

Solution

- 利用字典树进行单词的高速查找。
- 时间复杂度 $O(N^2)$ 。

I. Invoking the Magic

Shortest Judge Solution: 661 Bytes

Description

给定 n 对 pair , 保证这 $2n$ 个数一定由 n 个不同的数各出现恰好两次构成。

将这些 pair 分成若干组 , 使得每组内出现的数都在该组内出现恰好两次。

最小化最大的组包含的 pair 数。

- $n \leq 100000$ 。

Solution

- 建立一张 n 个点的无向图，每个点表示某个出现的数字。
- 对于 $\text{pair}(a, b)$ ，在 a 和 b 之间连一条边。
- 容易发现图中每个连通块都是环，答案即为环长的最大值（或连通块点数的最大值）。
- 时间复杂度 $O(n \log n)$ 。

F. Finding a Sample

Shortest Judge Solution: 1384 Bytes

Description

给定两个 N 维感知机，构造一个样本使得两个感知机对其的分类结果不同。

- $N \leq 200$ 。

Solution

- 若两个感知机完全一样，则无解。
- 否则枚举每一维 i ，解不等式组判断仅保留第 i 维和 bias 时是否有解。
- 时间复杂度 $O(N)$ 。

G. Gliding

Shortest Judge Solution: 940 Bytes

Description

从 $(s_x, s_y, 0)$ 飞行到 $(t_x, t_y, 0)$ ，横向速度大小不能超过 v_h ，
竖直下降速度大小为 v_p ，任意时刻需要满足 $z \geq 0$ 。

有 n 个出风口，在出风口 i 上方可以获得向上的速度 v_i ，出发点一定是出风口。

求从起点飞到终点的最短时间。

- $n \leq 4000$ 。

Solution

- 最优路线一定是在经过的每个出风口处上升一段距离再飞到下一个位置，使得飞过去最后一瞬间的高度 $z = 0$ 。
- 容易发现最优路线经过的出风口的向上速度 v_i 递增。
- 将所有出风口按照向上速度 v_i 从小到大排序。
- 动态规划，设 f_i 表示从起点飞到速度第 i 小的出风口的最短时间，枚举上一个出风口转移。
- 时间复杂度 $O(n^2)$ 。

E. Easy DP Problem

Shortest Judge Solution: 1096 Bytes

Description

给定一个 DP 的状态定义以及转移方程。

给定一个序列 a_1, a_2, \dots, a_n , q 次询问, 每次询问在 a 的某个区间跑 DP 时某个状态的 DP 值。

- $n, q \leq 100000$ 。

Solution

- 容易发现最终结果即为 a_l, a_{l+1}, \dots, a_r 的前 k 大值之和加上 $1^2 + 2^2 + \dots + (r - l + 1)^2$ 。
- 利用可持久线段树在 $O(\log n)$ 时间内查询区间前 k 大值之和。
- 时间复杂度 $O((n + q) \log n)$ 。

B. Bin Packing Problem

Shortest Judge Solution: 927 Bytes

Description

模拟 Bin Packing 问题的两个近似算法：

(1) FF：每次找到最靠左的可以放下该物品的背包。

(2) BF：每次找到剩余容量最小的可以放下该物品的背包。

■ $n \leq 10^6$ 。

Solution

- BF：使用 `std::set` 按剩余容量维护背包，每次在 `std::set` 里 `lower_bound`。
- FF：对背包按编号建立线段树，每个节点维护区间内背包剩余容量的最大值，每次查询时根据左儿子维护的值判断应该往左还是往右递归。
- 时间复杂度 $O(n \log n)$ 。

H. Huge Clouds

Shortest Judge Solution: 2252 Bytes

Description

给定 x 轴上方 n 个点光源以及 m 条线段作为障碍。

对于 x 轴上的某个位置 $(t, 0)$ ，如果存在一个点光源满足 $(t, 0)$ 到点光源连线不触碰任何障碍，则它被照亮。

需要统计未被照亮的部分的总长度。

- $n, m \leq 500$ 。

Solution

- 枚举一个点光源以及一条线段，则无法被该点光源照亮的部分是一个区间。
- 将区间拆成两种事件：
 - (1) 当 t 从 $l - \epsilon$ 增大至 l 时，接下来的位置里，点光源 i 被遮挡次数增加 1。
 - (2) 当 t 从 r 增大至 $r + \epsilon$ 时，接下来的位置里，点光源 i 被遮挡次数减少 1。

Solution

- 从左往右考虑每个事件，维护每个点光源被遮挡的次数以及被至少遮挡一次的点光源数量 cnt 。
- 若 $cnt = n$ ，则该事件到上一个事件中间的部分未被照亮。
- 时间复杂度 $O(nm \log(nm))$ 。

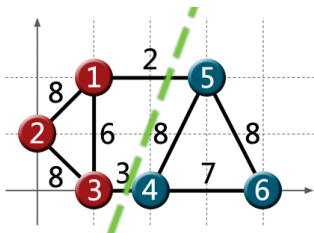
D. Dividing the Points

Shortest Judge Solution: 2199 Bytes

Description

给定平面上 n 个点以及对应的图。

找到一条不经过任何给定点的直线将 n 个点分成两个非空集合 A, B ，使得 $Ncut(A, B)$ 最小。



- $n \leq 1500$ 。
- $m \leq 5000$ 。

Solution

- 枚举两个点 P, Q , 考虑直线 PQ 稍微倾斜 ϵ 的情况即可考虑到所有情况。
- 枚举点 P , 将剩下的点 Q 以 P 为中心极角排序 , 转一圈的同时动态维护 $Ncut(A, B)$ 。
- 需要小心处理极角相同的情况。
- 时间复杂度 $O(n^2 \log n + nm)$ 。

L. List of Products

Shortest Judge Solution: 2188 Bytes

Description

定义两个正整数 A 和 B 的大小比较方式为：

按照 $2, 3, 5, 7, \dots$ 的顺序依次考虑每个质数 p ，如果 A 的质因数分解中 p 的次数不等于 B 中 p 的次数，则 p 的次数小的那个更小。

给定 $a[1..n]$ 和 $b[1..m]$ ，共 $n \times m$ 个 $a[i] \times b[j]$ ，求其中的第 k 小值。

- $n, m \leq 10^5$ 。
- $a_i, b_i \leq 10^6$ 。

Solution

- 不妨设 $n = m$, 所有数不超过 v 。
- 首先利用线性筛求出 $[2, v]$ 内每个数 i 的最小质因子 f_i , 沿着 f 一路除下去即可在 $O(\log v)$ 时间内完成质因数分解。

Solution

- 在 $n \times m$ 个 $a[i] \times b[j]$ 里二分答案，即找到最大的 $a[x] \times b[y]$ 满足至多有 k 对 $a[i] \times b[j]$ 不超过它。
- 给定 $a[x] \times b[y]$ ，需要统计有多少对 $a[i] \times b[j]$ 不超过它。
- 对序列 a 和 b 分别按题中定义的方式从小到大排序。
- 枚举 i ，双指针 j 即可，比较 $a[x] \times b[y]$ 与 $a[i] \times b[j]$ 的时间复杂度为 $O(\log v)$ 。

Solution

- 已知 $l < ans < r$, 如何得到二分需要检查的值 $a[x] \times b[y]$?
- 根据 $l < ans < r$, 对于每个 i 可以双指针求出满足 $l < a[i] \times b[j] < r$ 的 j 的范围 $[st_i, en_i]$ 。
- 在所有 $\sum en_i - st_i + 1$ 对 $a[i] \times b[j]$ 里随机选择一对作为当前二分需要检查的值 , 期望检查次数依然是 $O(\log n)$ 。
- 总时间复杂度 $O(n \log n \log v)$ 。

J. Just an Old Problem

Shortest Judge Solution: 2551 Bytes

Description

给定一个 n 个点 m 条边的连通无向图，求最小生成树的树边权值和以及方案数。

保证不存在经过 8 个点的简单路径。

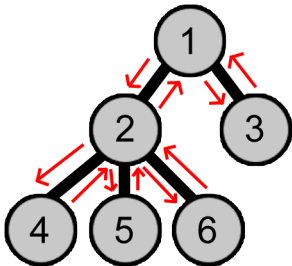
- $n \leq 50000$ 。
- $m \leq 100000$ 。

Solution

- 任取一棵输入图的 DFS 生成树 T 。
- 每条非树边 (u, v) 在 T 上互为祖孙关系。
- T 的深度不超过 7。

Solution

- 从根开始，按照 Euler Tour 进行动态规划。



- 从 1 开始的 Euler Tour 为

$1 \rightarrow 2 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 2 \rightarrow 6 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 1。$

Solution

- Euler Tour 中从 u 往下走过一条边 v 时：
- 遍历 v 往祖先连的所有非树边 (v, w) ，枚举每条非树边以及 (v, u) 是否加入 MST。
- 需要保证加边时两个点在 MST 中不连通。

Solution

- Euler Tour 中从 v 往上走过一条边回到 u 时：
- v 未来将不再有机会连出边。
- 需要保证 v 与 u 或 u 的祖先在 MST 中连通。

Solution

- 任意时刻，还能与未探索的部分连边的点只可能是当前走到点 x 及其祖先。
- 将 x 及其祖先最多共 7 个点的连通情况进行最小表示作为状态 S ，得到 DP 状态 $f_{i,S}$ 表示考虑了 Euler Tour 的前 i 步，连通情况为 S 时，MST 中加入的边的边权之和的最小值以及对应的方案数。
- 时间复杂度 $O((n+m)Bell(7))$ ，其中 $Bell(7) = 877$ 。

Thank you!