

第 15 届“连山科技”程序设计大赛

题目讲解

徐子懋

443756401@qq.com

BIT ACM Club

北京理工大学 ACM 俱乐部

November 29, 2020



编号	标题	难度	出题人	提交数	AC 数	通过率
I	Nobody Knows Better Than Me	简单	牟思睿	279	113	41%
A	IQ Difference	简单	龙水彬	207	112	54%
D	Life Game	简单	刘晒兮	152	98	64%
G	Nim Plus	中等	易翔宇	149	54	36%
L	Long Long Wanna Buy	中等	易翔宇	172	44	26%
M	Camouflage	中等	龙水彬	123	43	35%
J	Teacher Long and Machine Learning	中等	周赫彬	33	9	27%
C	Simple Anipop	中难	孙璋亮	11	3	27%
H	Treasure Hunt	中难	沈云汉	49	3	6%
K	Number Puzzle	中难	沈云汉	21	2	10%
E	Eigen Substring	困难	牟思睿	9	1	11%
B	Residue Problem	困难	谢威宇	1	0	0%
F	SVM	困难	沈云汉	0	0	0%

1. Easy

- 1.1. I - Nobody Knows Better Than Me
- 1.2. A - IQ difference
- 1.3. D - Life Game

2. Medium

- 2.1. G - Nim Plus
- 2.2. L - Long Long Wanna Buy
- 2.3. M - Camouflage
- 2.4. J - Teacher Long and Machine Learning

3. Medium Hard

- 3.1. C - Simple Anipop
- 3.2. H - Treasure Hunt
- 3.3. K - Number Puzzle

4. Hard

- 4.1. E - Eigen Substring
- 4.2. B - Residue Problem
- 4.3. F - SVM

1. Easy

1.1. I - Nobody Knows Better Than Me

1.2. A - IQ difference

1.3. D - Life Game

2. Medium

2.1. G - Nim Plus

2.2. L - Long Long Wanna Buy

2.3. M - Camouflage

2.4. J - Teacher Long and Machine Learning

3. Medium Hard

3.1. C - Simple Anipop

3.2. H - Treasure Hunt

3.3. K - Number Puzzle

4. Hard

4.1. E - Eigen Substring

4.2. B - Residue Problem

4.3. F - SVM



Problem

某国总统大选。一共有 n 个州，每个州有两个值 l 与 w 。懂王如果想要赢下一个州，必须在这个州上花费 $\max\{0, \lceil \frac{l-w}{2} \rceil\}$ 的时间。懂王现在有 t 个单位的自由时间，问懂王是否能赢？

- $1 \leq n \leq 2 \times 10^5, n \equiv 1 \pmod{2}$
- $0 \leq t \leq 2^{31}$
- $0 \leq l_i, w_i \leq 100, l_i + w_i < 100, l_i \neq w_i$

现场情况

首次提交：混子上分队 @ 00:11

首次通过：混子上分队 @ 00:13

通过队数：113



Problem

某国总统大选。一共有 n 个州，每个州有两个值 l 与 w 。懂王如果想要赢下一个州，必须在这个州上花费 $\max\{0, \lceil \frac{l-w}{2} \rceil\}$ 的时间。懂王现在有 t 个单位的自由时间，问懂王是否能赢？

- $1 \leq n \leq 2 \times 10^5, n \equiv 1 \pmod{2}$
- $0 \leq t \leq 2^{31}$
- $0 \leq l_i, w_i \leq 100, l_i + w_i < 100, l_i \neq w_i$

现场情况

首次提交：混子上分队 @ 00:11

首次通过：混子上分队 @ 00:13

通过队数：113

Solution

贪心地从开销最小的州开始计算能赢的州的数量即可。

将州按照 $\lceil \frac{l-w}{2} \rceil$ 进行递增排序，然后依次计算懂王是否能赢下当前的州。最后检查懂王能赢下的州的数量是否过半即可。

实现时需要特别注意边界情况的处理。

时间复杂度 $\mathcal{O}(n \cdot \log n)$ 。

1. Easy

1.1. I - Nobody Knows Better Than Me

1.2. A - IQ difference

1.3. D - Life Game

2. Medium

2.1. G - Nim Plus

2.2. L - Long Long Wanna Buy

2.3. M - Camouflage

2.4. J - Teacher Long and Machine Learning

3. Medium Hard

3.1. C - Simple Anipop

3.2. H - Treasure Hunt

3.3. K - Number Puzzle

4. Hard

4.1. E - Eigen Substring

4.2. B - Residue Problem

4.3. F - SVM

Problem

给出 n 位想参加 BITCPC 同学的智商 c_1, c_2, \dots, c_n , 3 人组队参赛, 且队伍内智商差不能超过 C , 询问最多能组成几只参赛队伍?

- $1 \leq n \leq 1\,000$
- $1 \leq C \leq 50$

现场情况

首次提交: 无中生 wa @ 00:04

首次通过: 无中生 wa @ 00:04

通过队数: 112

Problem

给出 n 位想参加 BITCPC 同学的智商 c_1, c_2, \dots, c_n , 3 人组队参赛, 且队伍内智商差不能超过 C , 询问最多能组成几只参赛队伍?

- $1 \leq n \leq 1\,000$
- $1 \leq C \leq 50$

现场情况

首次提交: 无中生 wa @ 00:04

首次通过: 无中生 wa @ 00:04

通过队数: 112



Solution

想要参赛队伍尽可能多，那么就贪心让智商相近的同学一块组队。

将同学按智商升序排序，若 i 号同学不能和 $i+1$ 、 $i+2$ 号同学组队，则他也肯定不能和后面其他同学组队（因为智商差距更大）。按顺序枚举三元组 $(i, i+1, i+2)$ ，判断他们能否满足组队条件，贪心地让能组队的连续三位同学组队即可。

时间复杂度 $\mathcal{O}(n \cdot \log n)$ 。

1. Easy

1.1. I - Nobody Knows Better Than Me

1.2. A - IQ difference

1.3. D - Life Game

2. Medium

2.1. G - Nim Plus

2.2. L - Long Long Wanna Buy

2.3. M - Camouflage

2.4. J - Teacher Long and Machine Learning

3. Medium Hard

3.1. C - Simple Anipop

3.2. H - Treasure Hunt

3.3. K - Number Puzzle

4. Hard

4.1. E - Eigen Substring

4.2. B - Residue Problem

4.3. F - SVM

Problem

给定由 $n \times m$ 个格子构成的矩阵，每个格子有两种状态，* 代表存活，. 代表死亡。在每个离散的时间节点，这些格子的状态都会同时发生变化，当格子周围 8 个格点中存活格点的数量在 $[l, r]$ 的范围内时，此格点会保持或变为存活状态，否则不会产生生命或死亡。

现给出矩阵的初始状态以及 l, r ，请你预测 t 时间后整个矩阵的状态。

- $1 \leq n, m \leq 100$
- $1 \leq t \leq 1000$

现场情况

首次提交: \zyw/ @ 00:15

首次通过: 这个队咋不起名字 @ 00:29

通过队数: 98

Problem

给定由 $n \times m$ 个格子构成的矩阵，每个格子有两种状态，* 代表存活，. 代表死亡。在每个离散的时间节点，这些格子的状态都会同时发生变化，当格子周围 8 个格点中存活格点的数量在 $[l, r]$ 的范围内时，此格点会保持或变为存活状态，否则不会产生生命或死亡。

现给出矩阵的初始状态以及 l, r ，请你预测 t 时间后整个矩阵的状态。

- $1 \leq n, m \leq 100$
- $1 \leq t \leq 1000$

现场情况

首次提交: \zyw/ @ 00:15

首次通过: 这个队咋不起名字 @ 00:29

通过队数: 98

Solution

本题只需按照题意模拟，通过前一时刻矩阵中微生物 8 个邻居的状态确定下一时刻生存状态的变化即可。

时间复杂度 $\mathcal{O}(8n \cdot m \cdot t)$ 。

1. Easy

- 1.1. I - Nobody Knows Better Than Me
- 1.2. A - IQ difference
- 1.3. D - Life Game

2. Medium

- 2.1. G - Nim Plus
- 2.2. L - Long Long Wanna Buy
- 2.3. M - Camouflage
- 2.4. J - Teacher Long and Machine Learning

3. Medium Hard

- 3.1. C - Simple Anipop
- 3.2. H - Treasure Hunt
- 3.3. K - Number Puzzle

4. Hard

- 4.1. E - Eigen Substring
- 4.2. B - Residue Problem
- 4.3. F - SVM

1. Easy

- 1.1. I - Nobody Knows Better Than Me
- 1.2. A - IQ difference
- 1.3. D - Life Game

2. Medium

- 2.1. G - Nim Plus
- 2.2. L - Long Long Wanna Buy
- 2.3. M - Camouflage
- 2.4. J - Teacher Long and Machine Learning

3. Medium Hard

- 3.1. C - Simple Anipop
- 3.2. H - Treasure Hunt
- 3.3. K - Number Puzzle

4. Hard

- 4.1. E - Eigen Substring
- 4.2. B - Residue Problem
- 4.3. F - SVM

Problem

有一堆 n 个石子，龙龙和懋懋各自只能取自己数集内的数字个数的石子。最后不能取的那个人失败，问先手必胜还是先手必败。

- $1 \leq n \leq 5000$
- $1 \leq m \leq 100$
- $1 \leq s_i \leq 500$

现场情况

首次提交：林克卡特大战动态树 @ 00:14

首次通过：林克卡特大战动态树 @ 00:14

通过队数：54

Problem

有一堆 n 个石子，龙龙和懋懋各自只能取自己数集内的数字个数的石子。最后不能取的那个人失败，问先手必胜还是先手必败。

- $1 \leq n \leq 5000$
- $1 \leq m \leq 100$
- $1 \leq s_i \leq 500$

现场情况

首次提交：林克卡特大战动态树 @ 00:14

首次通过：林克卡特大战动态树 @ 00:14

通过队数：54

Solution

本题是个博弈题，所以没有学过 SG 函数的同学可能需要先补充一下关于 SG 函数的知识。对于普通的 SG 函数，由于双方的操作方式是一致的，所以可以共享 SG 函数。但是对于本题，由于龙龙和懋懋的可移除的纸团数是不同的，所以需要给两个人分别构建 SG 函数。比如由龙龙的必败态可以通过懋懋的操作方式回推到懋懋的必胜态。然后同步更新双方的 SG 函数即可得到答案。但是赛时好像有很多出乎出题人意料的不是 SG 函数的解法导致过题队伍激增。时间复杂度 $\mathcal{O}(n \cdot m)$ 。

1. Easy

- 1.1. I - Nobody Knows Better Than Me
- 1.2. A - IQ difference
- 1.3. D - Life Game

2. Medium

- 2.1. G - Nim Plus
- 2.2. L - Long Long Wanna Buy**
- 2.3. M - Camouflage
- 2.4. J - Teacher Long and Machine Learning

3. Medium Hard

- 3.1. C - Simple Anipop
- 3.2. H - Treasure Hunt
- 3.3. K - Number Puzzle

4. Hard

- 4.1. E - Eigen Substring
- 4.2. B - Residue Problem
- 4.3. F - SVM

Problem

给出 n 种组件每种组件 m 个，每个组件有价值和使用寿命两种属性，在每种组件取一个的情况下，定义组件总价值除以组件的最小使用寿命的值越小，性价比越高，求性价比最高的情况下要花多少钱。

- $1 \leq n, m \leq 1000$
- $1 \leq s_i \leq 1000$
- $1 \leq p_i \leq 1000000$

现场情况

首次提交：我为吃狂 @ 00:18

首次通过：马拉车拉马 @ 00:25

通过队数：44

Problem

给出 n 种组件每种组件 m 个，每个组件有价值和使用寿命两种属性，在每种组件取一个的情况下，定义组件总价值除以组件的最小使用寿命的值越小，性价比越高，求性价比最高的情况下要花多少钱。

- $1 \leq n, m \leq 1000$
- $1 \leq s_i \leq 1000$
- $1 \leq p_i \leq 1000000$

现场情况

首次提交：我为吃狂 @ 00:18

首次通过：马拉车拉马 @ 00:25

通过队数：44

Solution

输入的组件是按照使用寿命排序过的，所以基本思路就是可以枚举最小的使用寿命，对每种组件按顺序逐个选购即可。但这里需要注意的是，寿命更长的组件可能更便宜，标程的做法是对每种组件倒过来预处理，如果寿命更长的组件更便宜，就直接覆盖前面的组件。

时间复杂度 $\mathcal{O}(n \cdot m)$ 。

1. Easy

- 1.1. I - Nobody Knows Better Than Me
- 1.2. A - IQ difference
- 1.3. D - Life Game

2. Medium

- 2.1. G - Nim Plus
- 2.2. L - Long Long Wanna Buy
- 2.3. M - Camouflage**
- 2.4. J - Teacher Long and Machine Learning

3. Medium Hard

- 3.1. C - Simple Anipop
- 3.2. H - Treasure Hunt
- 3.3. K - Number Puzzle

4. Hard

- 4.1. E - Eigen Substring
- 4.2. B - Residue Problem
- 4.3. F - SVM



Problem

给出 $n \times m$ 的像素图，其中有许多由 # 包围的。像素区域，要求填充面积小于 d 的像素区域。

- $1 \leq n, m \leq 1\,000$
- $1 \leq d \leq n \times m$

现场情况

首次提交：北湖施工 2 队 @ 00:19

首次通过：在座的各位 @ 01:05

通过队数：43

Problem

给出 $n \times m$ 的像素图，其中有许多由 # 包围的。像素区域，要求填充面积小于 d 的像素区域。

- $1 \leq n, m \leq 1\,000$
- $1 \leq d \leq n \times m$

现场情况

首次提交：北湖施工 2 队 @ 00:19

首次通过：在座的各位 @ 01:05

通过队数：43

Solution

整个地图规模只有 10^6 ，使用深度优先搜索（或者宽度优先搜索）计算每个由 # 包围的 . 像素区域的面积，对于面积小于 d 的区域再使用深度优先搜索（或者宽度优先搜索）填充一遍即可。注意，由于最外侧的 . 和由 # 所包围的内侧 . 是相同字符，可以先将最外层的 . 标记一下，防止枚举时候填到了最外层。

时间复杂度 $\mathcal{O}(n \cdot m)$ 。

1. Easy

- 1.1. I - Nobody Knows Better Than Me
- 1.2. A - IQ difference
- 1.3. D - Life Game

2. Medium

- 2.1. G - Nim Plus
- 2.2. L - Long Long Wanna Buy
- 2.3. M - Camouflage
- 2.4. J - Teacher Long and Machine Learning

3. Medium Hard

- 3.1. C - Simple Anipop
- 3.2. H - Treasure Hunt
- 3.3. K - Number Puzzle

4. Hard

- 4.1. E - Eigen Substring
- 4.2. B - Residue Problem
- 4.3. F - SVM



Problem

给 5 个点 f_1, f_2, f_3, f_4, f_5 , 找一个 4 阶的整系数且系数在 $[-100, 100]$ 范围内的多项式去拟合, 使得它在这五个位置和目标值的差不超过 1。

现场情况

首次提交: 蹭吃蹭喝蹭气球 @ 01:37

首次通过: YanheRedInWest @ 01:44

通过队数: 9



Problem

给 5 个点 f_1, f_2, f_3, f_4, f_5 , 找一个 4 阶的整系数且系数在 $[-100, 100]$ 范围内的多项式去拟合, 使得它在这五个位置和目标值的差不超过 1。

现场情况

首次提交: 蹭吃蹭喝蹭气球 @ 01:37

首次通过: YanheRedInWest @ 01:44

通过队数: 9



Part 1

注意到 $|F(x) - f_x| \leq 1$ ，而 $F(x), f_x$ 为整数，因此 $F(x)$ 的取值只有三种。

3^5 枚举 $F(x)$ 的每一种取值，即可消去不等式，4 阶多项式在已经有 5 个点的情况下可以直接插值求系数，然后判断是否符合条件即可。

$$f(x) = \sum_{i=1}^5 f_i \prod_{i \neq j} \frac{x-j}{i-j}$$



Part 2

不妨考虑暴力枚举高阶的系数，然后剪枝。

枚举 a_4, a_3, a_2 。由于系数的范围为 $[-100, 100]$ ，其中有很多项的偏差已经是剩余项补不回来的，这些情况都可以被提前丢掉。

此后原问题可转化为一阶方程在五个点上的问题。

$F(1) = a_1 + a_0$ ， $F(2) = 2a_1 + a_0$ ，因此已知 f_1, f_2 时 a_1 的取值最多只有五种，而 a_1 确定时 a_2 取值最多只有三种，可以直接枚举判定。

1. Easy

- 1.1. I - Nobody Knows Better Than Me
- 1.2. A - IQ difference
- 1.3. D - Life Game

2. Medium

- 2.1. G - Nim Plus
- 2.2. L - Long Long Wanna Buy
- 2.3. M - Camouflage
- 2.4. J - Teacher Long and Machine Learning

3. Medium Hard

- 3.1. C - Simple Anipop
- 3.2. H - Treasure Hunt
- 3.3. K - Number Puzzle

4. Hard

- 4.1. E - Eigen Substring
- 4.2. B - Residue Problem
- 4.3. F - SVM

1. Easy

- 1.1. I - Nobody Knows Better Than Me
- 1.2. A - IQ difference
- 1.3. D - Life Game

2. Medium

- 2.1. G - Nim Plus
- 2.2. L - Long Long Wanna Buy
- 2.3. M - Camouflage
- 2.4. J - Teacher Long and Machine Learning

3. Medium Hard

- 3.1. C - Simple Anipop
- 3.2. H - Treasure Hunt
- 3.3. K - Number Puzzle

4. Hard

- 4.1. E - Eigen Substring
- 4.2. B - Residue Problem
- 4.3. F - SVM



Problem

消去 n 个物体，当剩余的物体多于一个时，得到价值为消去的物体和相邻两个物体值的积。若只有一个物体，则得到的价值为这个物体值。

- $1 \leq n \leq 500$

现场情况

首次提交：林克卡特大战动态树 @ 02:04

首次通过：林克卡特大战动态树 @ 02:13

通过队数：3

Problem

消去 n 个物体，当剩余的物体多于一个时，得到价值为消去的物体和相邻两个物体值的积。若只有一个物体，则得到的价值为这个物体值。

- $1 \leq n \leq 500$

现场情况

首次提交：林克卡特大战动态树 @ 02:04

首次通过：林克卡特大战动态树 @ 02:13

通过队数：3

Part 1

- 方法: 区间 DP
- 很多情况下, 我们考虑环时, 可以先将题目简化为链
- 简化为链的方法通常就是, *copy & paste* 一次原区间, 例如将 $[1,2,3,4]$ 变为 $[1,2,3,4,1,2,3,4]$ 。我们发现在其中 $[1,2,3,4], [2,3,4,1], [3,4,1,2], [4,1,2,3]$ 这类环上的分割区间我们都能够找到。
- 可以发现, 消去最后一个物体的积分规则似乎和之前的规则没有很大的共性。
- 依然考虑简化一些的问题, 我们可以假设最后一个消去的是 x 。

Part 2

- 在这里我假设是上面 $[1,2,3,4]$ 中的 3。我们可以发现，我们需要通过消去规则 1 消去的区间是 $[4,1,2]$ 。是被两个 3 分割的区间。
- 同样可以发现，整个消去过程中，仍然存在特定的顺序关系，例如 4 绝对不会在 1 没有消去之前和 2 相邻，因为 3 是最后一个消去的物体，所以 4 不会越过 3 和 2 相邻。此时问题简化为了链。
- 对于 3 最后一个消去的情况（消去规则 2 获得的积分已确定），我们可以将问题转化为，消除 $[4,1,2]$ 使他们通过规则获得的积分最多。



Part 3

- $dp[i, j]$ 表示 i, j 位置存在未消去物体, 但 i, j 之间物体都被消去所获得的最大积分。
- 假设 i, j 之间最后一个被消去的物体位置是 k
- $dp[i, j] = \max(dp[i, k] + dp[k, j] + val[i] \cdot val[k] \cdot val[j])$
- 枚举 $j - i$ 长度, 枚举 i , 枚举 k 求 DP
- 转化成链后, 求最终答案 $ans = \max(val[i] + dp[i, i + n])$ 即可

时间复杂度 $\mathcal{O}(n^3)$ 。

1. Easy

- 1.1. I - Nobody Knows Better Than Me
- 1.2. A - IQ difference
- 1.3. D - Life Game

2. Medium

- 2.1. G - Nim Plus
- 2.2. L - Long Long Wanna Buy
- 2.3. M - Camouflage
- 2.4. J - Teacher Long and Machine Learning

3. Medium Hard

- 3.1. C - Simple Anipop
- 3.2. H - Treasure Hunt**
- 3.3. K - Number Puzzle

4. Hard

- 4.1. E - Eigen Substring
- 4.2. B - Residue Problem
- 4.3. F - SVM



Problem

给一个流量图，求所有点的 流量 \times 点权 和的最大值。

- 点数 ≤ 100
- 边数 ≤ 500
- 点权 ≤ 100

现场情况

首次提交：克鲁斯卡尔大战克苏鲁 @ 00:19

首次通过：dasjksstar @ 00:23

通过队数：3



Problem

给一个流量图，求所有点的 流量 \times 点权 和的最大值。

- 点数 ≤ 100
- 边数 ≤ 500
- 点权 ≤ 100

现场情况

首次提交：克鲁斯卡尔大战克苏鲁 @ 00:19

首次通过：dasjstar @ 00:23

通过队数：3

Solution

可以很容易的发现，这是一个费用流问题。

建图时可以将点拆分成两个点并用费用为点权的边连接，也可以将点的所有出边费用置为点权。

但是需要在这个费用流图上跑最小费用可行流，而不是最小费用最大流。

最小费用可行流只要在最小费用最大流的基础上稍作修改，当增广路费用大于零时就结束增广即可。

时间复杂度 $O(n \cdot m \cdot f)$ 。

1. Easy

- 1.1. I - Nobody Knows Better Than Me
- 1.2. A - IQ difference
- 1.3. D - Life Game

2. Medium

- 2.1. G - Nim Plus
- 2.2. L - Long Long Wanna Buy
- 2.3. M - Camouflage
- 2.4. J - Teacher Long and Machine Learning

3. Medium Hard

- 3.1. C - Simple Anipop
- 3.2. H - Treasure Hunt
- 3.3. K - Number Puzzle**

4. Hard

- 4.1. E - Eigen Substring
- 4.2. B - Residue Problem
- 4.3. F - SVM



Problem

给 $k = n \times m + 1$ 块拼图，保证第一列上的数是该列最小值且每个数都比他下一列的数小。请通过最多 $2k$ 次交互找到 $n \times m$ 的完整版拼图中多余的一块。

每次交互可以询问两个值是否在相邻的行或列中。

- $2 \leq n \leq m \leq 100$

现场情况

首次提交：\zyw/ @ 01:30

首次通过：\zyw/ @ 01:46

通过队数：2

Problem

给 $k = n \times m + 1$ 块拼图，保证第一列上的数是该列最小值且每个数都比他下一列的数小。请通过最多 $2k$ 次交互找到 $n \times m$ 的完整版拼图中多余的一块。

每次交互可以询问两个值是否在相邻的行或列中。

- $2 \leq n \leq m \leq 100$

现场情况

首次提交: \zyw/ @ 01:30

首次通过: \zyw/ @ 01:46

通过队数: 2

Part 1

解读原拼图满足的 2 条性质，发现：

- 每列第一个数是这一列中最小的数；
- 每列中的每个数都比上一列中每个数大；

因此我们可以将大体解题思路确定为：

- 确定 n 和 m 的值；
- 确定第一行的元素，并确定多余拼图所在的列；
- 确定多于拼图所在的列的元素，确定多余的拼图；

Part 2

将所有数字按照升序排序，记作 a_1, a_2, \dots, a_k 。

枚举 $k-1$ 所有 $\leq \sqrt{k-1}$ 的因子 f ，尝试将其作为 n 的值。若 a_1 与 a_{f+1} 水平相邻，或 a_1 与 a_{f+2} 水平相邻且 a_{f+2} 与 a_{2f+2} 水平相邻，则可确定 $n = f, m = \frac{k-1}{f}$ 。可以在 $2\sqrt{k}$ 次交互内完成。

以 n 为间隔确定第一行的元素与多余数字所在的列，多余数字所在的列一定包含 $n+1$ 个数字，因此若 a_{cn+1} 与 $a_{(c+1)n+1}$ 不相邻，则多余元素一定在第 c 列。可以在 m 次交互内完成。通过枚举确定多余数字所在列的顺序，最终得到多余的数字。可以在 n^2 次交互内完成。

总交互次数不会超过 $2\sqrt{k} + m + n^2 \leq 2k$ 次。

注意：需特判最小元素是否为多余数字。

1. Easy

- 1.1. I - Nobody Knows Better Than Me
- 1.2. A - IQ difference
- 1.3. D - Life Game

2. Medium

- 2.1. G - Nim Plus
- 2.2. L - Long Long Wanna Buy
- 2.3. M - Camouflage
- 2.4. J - Teacher Long and Machine Learning

3. Medium Hard

- 3.1. C - Simple Anipop
- 3.2. H - Treasure Hunt
- 3.3. K - Number Puzzle

4. Hard

- 4.1. E - Eigen Substring
- 4.2. B - Residue Problem
- 4.3. F - SVM

1. Easy

- 1.1. I - Nobody Knows Better Than Me
- 1.2. A - IQ difference
- 1.3. D - Life Game

2. Medium

- 2.1. G - Nim Plus
- 2.2. L - Long Long Wanna Buy
- 2.3. M - Camouflage
- 2.4. J - Teacher Long and Machine Learning

3. Medium Hard

- 3.1. C - Simple Anipop
- 3.2. H - Treasure Hunt
- 3.3. K - Number Puzzle

4. Hard

- 4.1. E - Eigen Substring
- 4.2. B - Residue Problem
- 4.3. F - SVM

Problem

定义母串 s 的某个子串 $s[l..r]$ 是其“特征子串”当且仅当该子串仅在 s 中出现一次。
现给定长度为 n 的母串 s 。分别求母串 s 的每个前缀的最短特征子串长度。

- $1 \leq n \leq 10^6$
- $|\Sigma| = 26$

现场情况

首次提交: FG204 @ 01:21
首次通过: dasjstar @ 03:41
通过队数: 1

Problem

定义母串 s 的某个子串 $s[l..r]$ 是其“特征子串”当且仅当该子串仅在 s 中出现一次。
现给定长度为 n 的母串 s 。分别求母串 s 的每个前缀的最短特征子串长度。

- $1 \leq n \leq 10^6$
- $|\Sigma| = 26$

现场情况

首次提交: FG204 @ 01:21
首次通过: dasjstar @ 03:41
通过队数: 1

Part 1

子串枚举

- 考虑对母串 s 建立后缀自动机。
- 在后缀自动机上沿母串 s 进行转移。
- 维护一个多重集 S ，表示转移过程中发现的所有子串。
- 每转移到一个节点 t ，将 t 沿 fail 链直到自动机起始节点路径上所有节点所表示的子串加入到 S 中。
- 每转移到一个节点，输出 S 中仅出现一次且最短的串的长度。

Part 2

优化实现

- 可以使用优先队列维护多重集 S 。
- 为后缀自动机上每个节点维护一个“访问次数”值 w 。
- 每转移到一个节点 t ，从 t 开始，暴力沿 fail 链向自动机起始节点枚举。
- 每枚举到一个节点 t' ，判断其 w 值。若 $w = 0$ ，则将 w 递增并将 t' 加入到优先队列中；若 $w = 1$ ，则将 w 递增并继续枚举；若 $w = 2$ ，则停止枚举。
- 优先队列的排序依据为后缀自动机节点的 fail 节点的 len 值。
- 当输出答案时，首先不断弹出位于优先队列队头的、 w 值等于 2 的节点；然后输出队头节点的 fail 节点的 $len + 1$ 作为答案。
- 时间复杂度 $\mathcal{O}(n \cdot \log n)$ 。

1. Easy

- 1.1. I - Nobody Knows Better Than Me
- 1.2. A - IQ difference
- 1.3. D - Life Game

2. Medium

- 2.1. G - Nim Plus
- 2.2. L - Long Long Wanna Buy
- 2.3. M - Camouflage
- 2.4. J - Teacher Long and Machine Learning

3. Medium Hard

- 3.1. C - Simple Anipop
- 3.2. H - Treasure Hunt
- 3.3. K - Number Puzzle

4. Hard

- 4.1. E - Eigen Substring
- 4.2. B - Residue Problem**
- 4.3. F - SVM

Problem

使函数 $f(i, r, P)$ 等于所有满足方程 $a^r(b + b^2)^i \equiv b^i \pmod{P}$ 的 $(a, b), 0 \leq a, b < P$ 有序对的个数。给出 m 个询问，每个询问有 Y, N, r, P ， P 一定是质数，回答 $\prod_{i=1}^N Y^{f(i, r, P)} \pmod{(10^9 + 7)}$ 的值。

现场情况

首次提交：为什么你们这么熟练啊 @ 03:24

首次通过：无

通过队数：0

Problem

使函数 $f(i, r, P)$ 等于所有满足方程 $a^r(b + b^2)^i \equiv b^i \pmod{P}$ 的 $(a, b), 0 \leq a, b < P$ 有序对的个数。给出 m 个询问，每个询问有 Y, N, r, P ， P 一定是质数，回答 $\prod_{i=1}^N Y^{f(i, r, P)} \pmod{(10^9 + 7)}$ 的值。

现场情况

首次提交：为什么你们这么熟练啊 @ 03:24
首次通过：无
通过队数：0

Part 1

注意：以下讨论中的运算都是模 P 域中的运算。

解 $f(i, r, P)$ $a^r(b + b^2)^i = b^i$ 经变换可得 $a^r(1 + b)^i b^i = b^i$ ，当 $b = 0$ 时，任意 $a \in [0, P)$ 符合要求，有 P 个解。当 $b \neq 0$ 时，化简得 $a^r(1 + b)^i = 1$ 。当 $b = P - 1$ 时，无解。

接下来对此式在 $b \in [1, P - 1)$ 情况下进行讨论。

Part 2

当 $r = 1$ 时, 对于任意的 b 都有唯一解 $a = (1 + b)^{-i}$ 。共有 $2P - 2$ 个解。

当 $r = 2$ 时:

如果 i 为奇数, 令 $i = 2k + 1$, 有 $(a(1 + b)^k)^2 = (1 + b)^{-1}$ 。因为 $b \in [1, P - 1)$, 故 $b + 1 \in [2, P)$, 且由于 1 的逆元就是 1, 所以 $(b + 1)^{-1} \in [2, P)$ 。由于 1 是一个二次剩余, 所以 $[2, P)$ 中有 $\frac{P-1}{2} - 1$ 个二次剩余。对每一个二次剩余 x , 存在两个 a 使得 $a^2 = x$ 。故总共有 $P + 2(\frac{P-1}{2} - 1) = 2P - 3$ 个解。

如果 i 为偶数, 令 $i = 2k$, 有 $a^2 = ((1 + b)^{-k})^2$ 。对于任意 $b \in [1, P - 1)$, $((1 + b)^{-k})^2$ 都是一个二次剩余, 所以都有两个解。故总共有 $P + 2(P - 2) = 3P - 4$ 个解。

Part 3

当 $r = 3$ 时, $a^3 = (1 + b)^{-i}$ 。

当 $P = 3t + 2$ 时, 对于任意 $x \in [1, P)$, $a^3 = x$ 都有且只有一个解。推导如下: 由 $a^3 = x$ 可得 $x^{\frac{P-2}{3}} = a^{P-2} = a^{-1}$, 所以这个解为 $a = x^{-\frac{P-2}{3}}$ 。所以, 对于每个 b , $a^3 = (1 + b)^{-i}$, 只有一个解。故总共有 $P + P - 2 = 2P - 2$ 个解。

当 $P = 3t + 1$ 时, $[2, P - 1)$ 中有 t 个三次剩余。

当 $i = 3k$ 时, 有 $a^3 = ((1 + b)^{-k})^3$, 同理有 $P + 3(P - 2) = 4P - 6$ 个解。

当 $i = 3k + 1$ 时, 有 $(a(1 + b)^k)^3 = (1 + b)^{-1}$, 同二次剩余的情况, 有 $P + 3(\frac{P-1}{3} - 1) = 2P - 4$ 个解。

当 $i = 3k + 2$ 时, 有 $(a(1 + b)^{k+1})^3 = (1 + b)$ 。同理有 $P + 3(\frac{P-1}{3} - 1) = 2P - 4$ 个解。

Part 4

计算 $\prod_{i=1}^N Y^{f(i,r,P)} \bmod (1e9+7)$: 变换得 $Y^{\sum_{i=1}^N f(i,r,P)} \bmod (1e9+7)$, 令

$Z = \sum_{i=1}^N f(i,r,P)$, 由欧拉降幂转化为 $Y^{(Z \bmod 1e9+6)+1e9+6}$, 计算即为最终答案。

1. Easy

- 1.1. I - Nobody Knows Better Than Me
- 1.2. A - IQ difference
- 1.3. D - Life Game

2. Medium

- 2.1. G - Nim Plus
- 2.2. L - Long Long Wanna Buy
- 2.3. M - Camouflage
- 2.4. J - Teacher Long and Machine Learning

3. Medium Hard

- 3.1. C - Simple Anipop
- 3.2. H - Treasure Hunt
- 3.3. K - Number Puzzle

4. Hard

- 4.1. E - Eigen Substring
- 4.2. B - Residue Problem
- 4.3. F - SVM

Problem

给 2 个 3 维点集，问这 2 个点集是否线性可分，若是则求出点集到分割平面的距离。

现场情况

首次提交：无

首次通过：无

通过队数：0

Problem

给 2 个 3 维点集，问这 2 个点集是否线性可分，若是则求出点集到分割平面的距离。

现场情况

首次提交：无
首次通过：无
通过队数：0

Part 1

首先分别求 2 个点集的三维凸包，使用增量法或者卷铺盖法都可以，时间复杂度分别为 $O(n^2)$ 和 $O(n^{1.5})$ ，实际运行效率差不多。

判断 2 个凸包是否有交，若有交则线性不可分，输出-1，由于凸包的面数不会超过 2 倍凸包的顶点数，因此时间复杂度为 $O(n^2)$ 。

三维情况和二维类似，凸包有交的充要条件为其中一个凸包的顶点在另一个凸包中，或两凸包的面有交。

Part 2

两凸包间的距离只有可能在下面 4 种情况中取得：

- 点到点的距离；
- 点到棱的距离；
- 点到面的距离；
- 棱到棱的距离；

时间复杂度为 $O(n^2)$ 。

因此总时间复杂度也为 $O(n^2)$ 。

Part 3

需要注意的细节：

点集中点的数量小于 3 个时，没法形成凸包，需要当作单独的点或棱进行特判；

点集中的点可能共线或共面，在求凸包时需要加入随机噪声，使得能够形成非退化的凸包；

在计算点到棱的距离时，需要判断点在棱所在直线的投影是否在棱上；

在计算点到面的距离时，需要判断点在面上的投影是否在边界范围内；

在计算棱到棱的距离时，需要判断两条棱所在直线中垂线的垂足是否分别在棱上。

各种地方精度的处理。

- 感谢连山科技的大力支持；
- 感谢学校以及学院对赛事的大力支持；
- 感谢林永钢老师的大力支持；
- 感谢出题人团队对赛题的辛苦准备；
- 感谢验题人团队对赛题的充分验证；
- 感谢赛场志愿者的辛勤付出。