

2021 Wuhan University Freshman Programming Contest

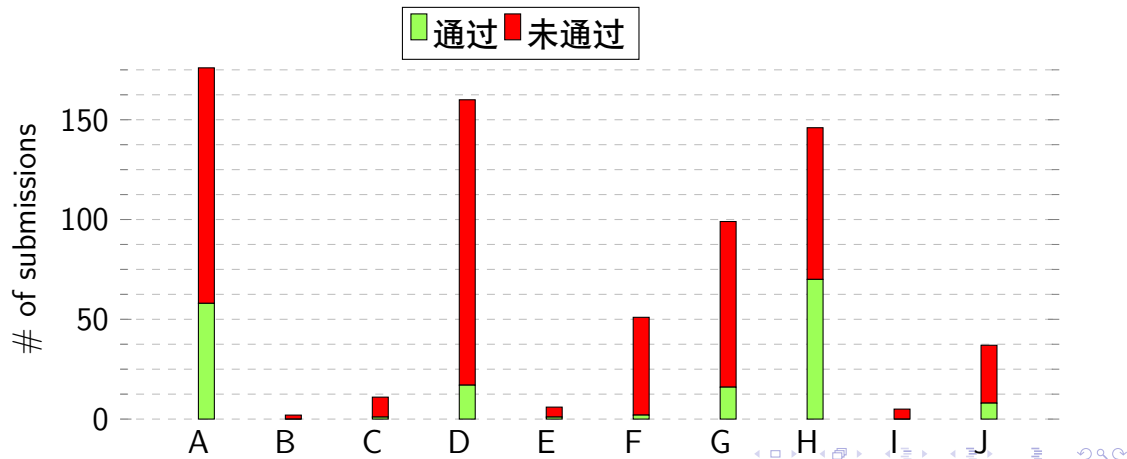
WHU ICPC 集训队

Apr. 4th, 2022

比赛小结

- 本次比赛共收到 777 份提交代码。
- 其中 185 份代码正确。
- 82 名参赛选手有提交记录。
- 73 名参赛选手至少通过一题。

各题通过情况



A. 仓鼠快速签到

解法

前 7 道题的限制唯一确定了全部 10 道题的答案，因此选手完全不需要知道后 3 题的答案。

对于完全不会写代码的萌新，这题可以手撕。可以从限制性比较强的题目入手，用常识也可以排除后 3 题的部分选项，请不要放弃。

对于会写 for 和 if 的选手，枚举每道题的答案共 3^{10} 种情况，再对前 7 题的限制进行验证即可。代码可以写的很短。

如果你不会写第 7 题的限制，前 6 道题限制后也只有 5 种情况，再手动验证第 7 题即可。

B. 二维弹球

题意

计算几何入门模拟题。

碰撞时，沿边的方向速度不变，垂直于边方向的速度反向。

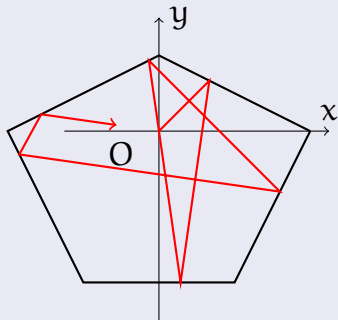
给定初始位置、速度向量、时间，求最终位置。

出题人的话

计算几何好难出，凸包生成了很多 $n = 20$ 的包，只能手动构造 $n = 100$ 的。希望大家善待计算几何。

B. 二维弹球

样例 2 的图示



B. 二维弹球

题解

根据题目描述，进行模拟。

注意到 $t \leq 100$ ，可按时间模拟。但可能一秒钟出现多次碰撞，判断起来较为麻烦。

注意到碰撞次数 $\leq 10^4$ ，按碰撞模拟。

常用的“点 + 向量 = 点”技巧，球当前位置是一个点，速度向量与时间做数乘是下一个位置。可以用当前位置与下一个位置的连线所在的直线判断下一个碰撞点，这时需要用到的知识点是直线交点。

B. 二维弹球

题解

碰撞时，沿边的方向速度不变，垂直于边方向的速度反向。

那么这时需要旋转速度向量。那么旋转多少度呢？求出速度向量与边向量的夹角 θ ，逆时针旋转 2θ 即可。可以证明，无论 θ 是锐角还是钝角，都是正确的。

所以我们只需求出每一次碰撞的位置，算出碰撞间隔时间，就可以找到圆最终的位置。

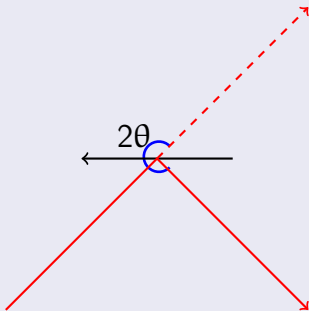
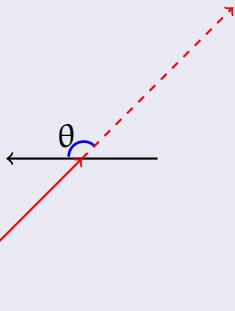
B. 二维弹球

锐角的情况



B. 二维弹球

钝角的情况



B. 二维弹球

题解

因此对一个向量 (x, y) ，逆时针旋转 θ ，所得到的新向量为 $(x \cos \theta - y \sin \theta, y \cos \theta + x \sin \theta)$ 。

求两线段交点比较容易，用面积法和向量去找。

总的时间复杂度为 $O(n \text{ 碰撞次数})$ 。

题意

有 N 个水果，第 i 个水果会于 t_i 时刻末在位置 p_i 落下，如果玩家在此时恰好移动到同一位置，就接住了这个水果。

有以下 3 种移动方式，分别是原地不动，移动一个位置，或等概率移动 1 个位置到 k 个位置。

求在最优的移动策略下，接到水果个数的期望。

解法

注意到要求解的是接住水果个数的期望，因此我们需要倒序对水果进行 dp。

设 $dp(i, j)$ 表示时刻 i 初，玩家处在位置 j ，后续能接住个数的最大期望值。

列出 3 种转移方式，从时刻 $i + 1$ 向时刻 i 转移。

$$dp(i, j) = v(i, j) + \max \left\{ \begin{array}{l} dp(i + 1, j), dp(i + 1, j - 1), dp(i + 1, j + 1) \\ \frac{1}{k} \sum_{d=j-k}^{j-1} dp(i + 1, d), \frac{1}{k} \sum_{d=j+1}^{j+k} dp(i + 1, d) \end{array} \right.$$

其中， $v(i, j)$ 表示 i 时间 j 位置是否落下水果。

区间和可以用前缀和优化，注意特判快速移动遇到边界的情况。

最后答案为 $\max_{1 \leq j \leq m} dp(1, j)$

这种做法的复杂度为 $O(m * t_n)$ 。

D. 和谐之树

题意

求对区间 $[1, n]$ 建立线段树后最大节点编号

D. 和谐之树

题解

考虑最大节点编号在当前左子树还是右子树。

如果左边比右边深度更深，那么在左子树，否则在右子树。

可以发现，当区间长度从 2^x 变到 $2^x + 1$ 长度时，线段树深度会 $+1$ 。

快速比较两个子树深度即可

E. 和谐之树 · 改

题意

区间询问 D 题的最大编号之和。

E. 和谐之树 · 改

题解

考虑选择左子树的时候，左边区间长度必须是 $2^x + 1$ ，右边长度必须是 2^x ，那么选择左子树时，当前区间的二进制必须形如 $100\dots001$ 。

考虑从当前长度 n 的节点走到某棵子树时，会变成 $n \gg 1 \mid (1 \text{ or } 0)$ ，那么走左子树的过程最多只会发生一次，即最开始长度的二进制表示中，第二高位 1 的位置处。

进一步发现，区间长度 n 最高位的两个 1 确定后，最大编号也确定了。那么 $1e18$ 内的不同最大编号只有 $\log^2 v$ 级别。打表同样也能发现。预处理后即可回答。注意 $T \log^2 v$ 的 \log 比较大，可能不能通过。

F. 仓鼠与炸弹

题意

问字符串 S 有多少个字串 $T = S[l, r]$ 满足：

1. T 的长度大于等于 $2m$
2. $T[1, m] = \text{reverse}(T[|T| - m + 1, |T|])$

F. 仓鼠与炸弹

题意

问字符串 S 有多少个字串 $T = S[l, r]$ 满足：

1. T 的长度大于等于 $2m$
2. $T[1, m] = \text{reverse}(T[|T| - m + 1, |T|])$

解法

字符串哈希：

定义：字符串 $S = s_1, s_2, \dots, s_n$ 的哈希为

$$\text{Hash}(S) = (s_1a^1 + s_2a^2 + \dots + s_na^n) \pmod{p}$$

我们认为，如果两个字符串的 hash 值相等，那么他们大概率是相等的
同时，我们可以 $O(1)$ 地求出 $\text{Hash}(S[l, r])$ 和 $\text{Hash}(\text{reverse}(S[l, r]))$

F. 仓鼠与炸弹

解法

从左到右枚举 i , 找到有多少个位置 $j \leq i - m$ 满足

$\text{Hash}(\text{reverse}(S[i - m + 1, i])) = \text{Hash}(S[j - m + 1, j])$ 复杂度 $O(n)$ 或者 $O(n \log n)$

细节

如果只对一个 10^9 级别的质数取模的话, 大概率会 WA (生日悖论)

如果对 2^{64} 次方自然溢出取模的话, 可以构造出两个不同的长度在 1000 左右的字符串, 他们的哈希值相同

G. 寄寄子的生日

题意

给 $2 \sim n$ 排列，只能交换值互质的两个数，构造一个 $2n$ 次交换后有序的方案。

G. 寄寄子的生日

题解

考虑通过找到最大的质数作为中间点交换，可以在两步内将任意一个数归位。

具体的 $a\ b\ p \rightarrow p\ b\ a \rightarrow b\ p\ a$ ，假如 p 是最大质数，那么可以把 b 归位。

关于 $2 \sim n$ 内最大的质数 p ，一定有 $2 \times p > n$ 成立，至少在题目范围内是正确的。

由于 n 开的比较小，因此直接暴力也可以通过，主要是想多给大家一点签到题。

题意

构造一个长为 n 的字符串，使子序列 `whu` 个数最多。

题解

题目名已经把做法写出来了，显然 $www...hhh...uuu...$ 的相应子序列个数最多。

注意 w, h, u 谁多谁少并不重要，只要三者个数差小于等于 1 即可。

证明可以考虑调整法，如果不是 $www...hhh...uuu...$ 往这个方向调整不会变差，如果个数差大于 1 往小于 1 调整答案也不会变得更差。

I. 异度之刃

题意

线段树套路题

不带修改的区间本质不同连续上升子串

题解

对于不带修改的题目，可以考虑将询问离线，移动右端点，每次移动后维护左端点的答案

I. 异度之刃

题解

那么对于本题，我们假设当前的处理右端点为 i 的询问，而线段树上查询 $[j, i]$ 的区间和表示以 j 为左端点的区间的答案

从 $i-1$ 到 i 时，需要修改线段树上一部分的位置，我们来考虑最右边新加一个点会产生什么影响

设 $g[i]$ 表示以 i 为右端点的最长连续上升子串的长度，显然

$$g[i] = \begin{cases} g[i-1] + 1 & a[i] = a[i-1] + 1 \\ 1 & \text{otherwise} \end{cases}$$

那么首先 $[i - g[i] + 1, i]$ 这一段区间进行区间 $+1$ 。因为这些以 i 为右端点的连续上升的子串都是因为这个新的点而出现的

I. 异度之刃

题解

但是题面要求本质不同，这些串可能在前面出现过，需要把他们在线段树上“删掉”，所以需要一些栈去维护这些旧串的位置

具体来说就是对于每一种右端点的值 c 开一个栈（可以用链表实现），栈上的每一个元素有两个信息，分别为旧串出现位置以及它的长度区间（其实是三个元素，不过长度区间的上/下限可以通过前一个出栈的元素的长度上/下限推断出来）。

现在我们需要加入一个新的元素入栈，他的位置为 i ，长度为 $1 \sim g[i]$ 。如果当前栈顶元素的长度比 $g[i]$ 小，那就直接出栈且完全在线段树的对应位置上删除，并且重复这个过程。如果比 $g[i]$ 大，那么就不用出栈，同时在线段树上删除被重复的部分。最后再将这个新的元素压入栈顶
维护完毕后在线段树上查询需要的区间即可

J. 传闻档案

题意

给一个带权有向图，每个点价值为此点能够到达的点的最大权值，求所有点价值和。

J. 传闻档案

解法

反向建边，按点权从大到小进行 dfs 即可。每个点的价值即为这次 dfs 根节点的权值。

注意每个点只访问一次。

也可以使用 tarjan 缩点，每个连通块的权值变为块中点的最大权值，缩点后 dfs 求出答案。