

USSTSIW-ACM集训队选拔赛

Easy-: B, C, M

Easy: A, F, G

Easy+: J, L, E

Medium: D, H

Medium+: I, K

A. Haruhi with Acem-hcpc

First solved: 杨欣悦, 02:30 (+2).

怎么模拟题也能写锅啊我好菜啊。

出数据的时候出锅了，导致会出现同一时间同一队交了同一题得到了 WA 和 AC 各一发的情况，稳定排序和不稳定排序出来的结果可能会导致罚时有 20 分钟的出入。只要其他写对，比赛的时候都认为是过了。没过的同学都有一些奇奇怪怪的错误。

以下是过这题需要注意的一些细节。

- 对封榜后的提交时间排一次序。
- 对已经 AC 的题的再次提交不产生任何影响。
- 注意排名的条件。
- 读入。

这题放到 OJ 上后可能会略微修改题面，比赛时能过的代码不一定能过了。

B. Haruhi with Bubbles

First solved: 王必成, 00:11 (+).

First solved(Rejudge前): 杨欣悦, 00:19 (+2).

过于简单，然后出锅。

注意每一秒钟每个泡泡只会进行一次操作：合并或爆掉或维持不变。

C. Haruhi with Changes

First solved: 李扬, 00:08 (+).

C 题主要考察的是四联通区域的基本操作，注意数据的范围，不能直接进行模拟。细心的同学其实非常容易发现：对于任何一个矩阵元素来说，如果该元素进行偶数次的操作等于没有任何操作；而对于其奇数次操作相当于进行了一次翻转操作，题目只要对每一个操作数矩阵的元素值进行奇偶判断即可，如果为奇数则对应元素及其四联通的连接点全部变化一次，如果为偶数则不进行任何操作，最后输出结果即可。

D. Haruhi with "Day of sagittarius"

究极简化自今年西安邀请赛的 H 题。

题意可以简化成，给一棵全白的树，每次挑一个点染成黑色，求白点的联通块数。

乍一看题目好像要让你实现一个能删除的并查集。然而这样基本写不出来。

不妨逆向思维一下，染黑的逆操作是染白，我们发现染白的操作就是并查集裸题。于是只要把所有操作离线了，倒着跑一遍存下答案这题就做完了。

PS：为啥很多同学都认为这题强制在线鸭，在线题不是这样出的鸭QAQ。

E. Haruhi with Equilibrium game

First solved: 王必成, 02:51 (+2).

华理原题再放送。

如果只有一堆石子，显然先手可以全部拿完取得胜利，如果拿剩下 ≥ 2 个，后手相当于面对着相同的局面，如果先手拿剩下 1 个，后手就只能乖乖拿完这一个，如果有下一堆，主动权仍然在先手手中。因此每一堆石子本质上只存在 1 个/多个 2 种状态，遇到只有 1 个的堆，别无选择。遇到有多个的堆，有两种选择，拿完or拿剩 1 个，并且这两种状态的胜败必然是反的，因此谁先遇到多个的石子堆就有必胜选法。因此每一轮的游戏，只看谁先能遇到 ≥ 2 的堆。亦即，从第 i 堆开始的胜负情况，只需要从第 i 堆往后走，看第一个非 1 的堆是谁拿。当然需要一点优化，不然会被卡。

F. Haruhi with Fish

First solved: 李扬, 01:32 (+1).

递推:

直接打个表很容易找到规律，很容易看出 $f_n = f_{n-1} + f_{n-3} + f_{n-4}$ ，注意过程中的取模。

或者分析:

题目要求不含 OoO 和 OOO 的串。

设 f_n 为长度为 n 串符合条件的个数。

则很明显在长度为 $n - 1$ 并且符合条件的串后面加上一个 o 一定符合。

如果在长度为 $n - 1$ 的串后面加上一个 O 的话我们得考虑 $n - 1$ 的串结尾的元素。

如果是 oo 的话我们可以看做是长度为 $n-3$ 的串加上 Ooo ？。

如果是 Oo 的话我们可以看做长度为 $n - 4$ 的串加上 $OOoo$ 。

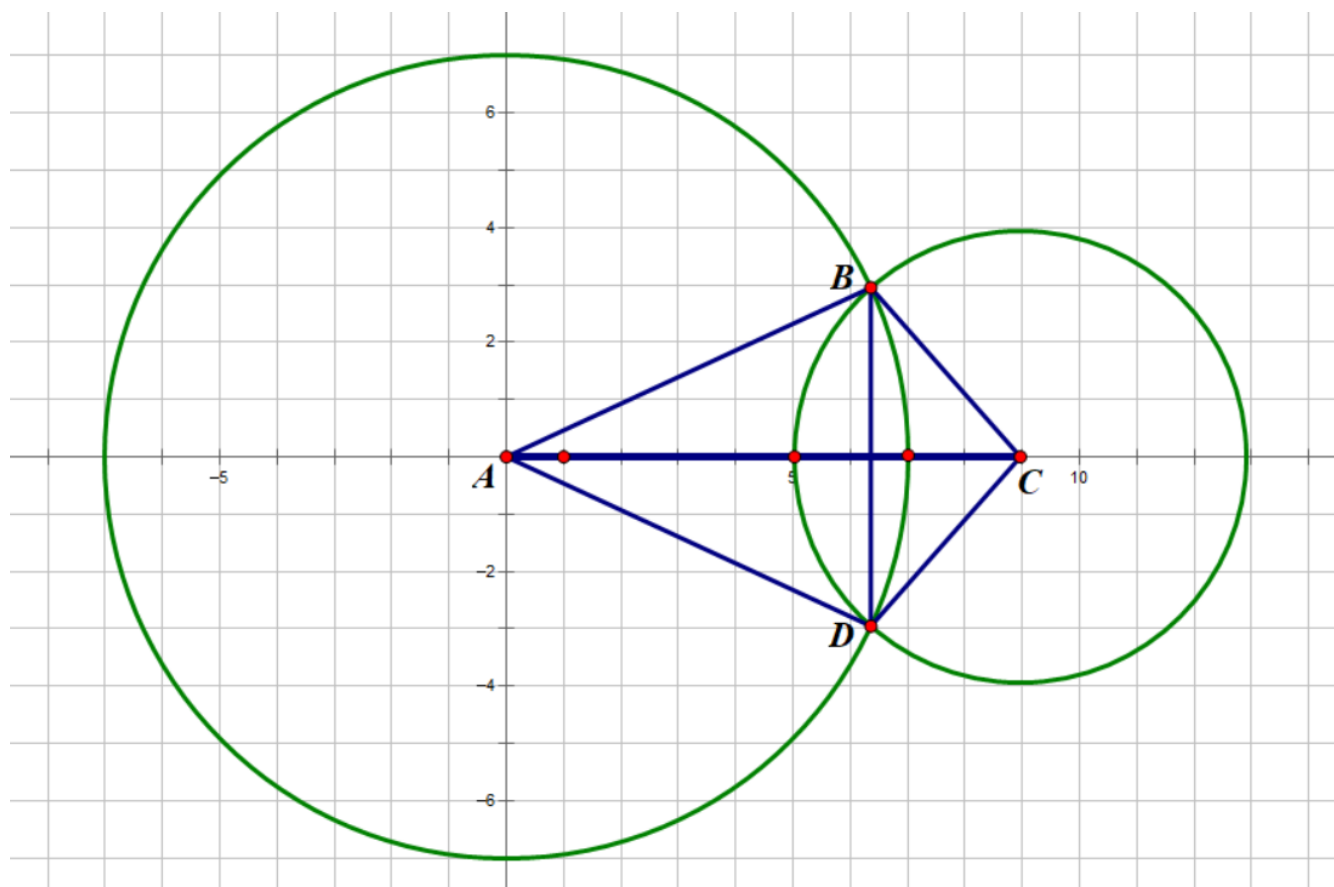
因此 $f_n = f_{n-1} + f_{n-3} + f_{n-4}$ 。

G. Haruhi with Global offensive

First solved: 诸葛文, 04:20 (+5).

首先这个概率就是两圆重合面积除以射击面积。

然后这题就变成了只靠高中数学就能做的零门槛计算几何题。



首先分包含，相离，相交三种情况讨论。前两种很显然，讲一下相交的情况。

可以发现重叠的面积就是扇形 ABD 加上扇形 CBD 减去四边形 $ABCD$ 。

由于三角形 ABC 三边都已知，面积很好求，四边形 $ABCD$ 就是两倍的三角形 ABC 。

考虑求两个扇形的面积，发现其圆心角的一半就是三角形 ABC 的两个角 $\angle CAB$ 和 $\angle ACB$ ，余弦定理搞一搞。

坑点是求这两个角的时候不能用反正弦求，因为这两个角的其中一个有可能是钝角。

H. Haruhi with Haruhi with Haruhi

五维偏序，对某个 *haruhi* 来说，全面碾压她的 *haruhi* 个数，是五个集合的交集的大小，这五个集合分别是单个维度比她强的 *haruhi*。

对于每个 *haruhi*，能快速得出这五个集合，相与就可以得到答案了。对于每一维按照第 1 名到最后 1 名的顺序以哈希形式预处理该维度比她强的人的 *id*，即维护一个 $b_{x,y}$ ，若为 1 则表示在该维度第 x 名的 *haruhi* 比第 y 号 *haruhi* 强(注意名次和编号的区别...)。遗憾的是这样做的复杂度和暴力没有任何区别，并且内存消耗很大，但是可以使用 *bitset* 加速，从而将时间消耗和内存消耗都减少 32/64 倍，已经足够通过本题。

上面做法中维护了整整 $5 * n$ 个大小为 n 的 *bitset*，但是同维度相邻的两个 *bitset* 只差 1 位而已(第 $x + 1$ 名的 *bitset* 和第 x 名的区别只有第 x 名这个人被标记了)，可以考虑少维护一些，比如每隔 100 名维护一个 *bitset*，100 个人共用这一份，然后对误差涉及的少数人暴力检查得出正确答案。由基本不等式可以算出来，最佳分块大小在 \sqrt{n} 左右，可以达到的最终复杂度为 $5 * n * \sqrt{n}$ 。

I. Haruhi with Inconsistent rain

这题本质上是 21 - F 的究极强化版。

本来想给补过 44 - H 的同学送分，然而搞成送命了QAQ。

设 $D(i, j)$ 是 i 结点与 j 结点的树上距离，答案显然是 $2 * \min_{1 \leq i \leq n} \{ \sum_{j=1}^n e_j \times D(i, j) \}$ ，等价于求树的带权重心。

先把 1 看做根，*dfs* 一遍求出以 1 作为整棵树的根时，以 i 为根的子树中所有 e_j 之和 f_i 、选 1 为仓库时的答案 dp_1 还有整棵树的 e_i 之和 sum 。

再做一次 *dfs* 开始换根，考虑把根 x 换为根 y 时答案的变化，所有 y 子树上的结点对答案的贡献都会减少 $f_y \times D(x, y)$ ，除去 y 子树上的结点的贡献都会增加 $(sum - f_y) \times D(x, y)$ 。

于是得到

$$dp_y = dp_x + (sum - f_y) \times D(x, y) - f_y \times D(x, y) = dp_x + (sum - 2 \times f_y) \times D(x, y)$$

更新答案就完事啦。

J. Haruhi with Jenga

First solved: 何智威, 04:54 (+).

图相当有提示性，本身 $1 - n$ 的纸条，就是 1 和 2 连接，2 和 3 连接... $n - 1$ 和 n 连接，一番折叠后，这些连接依然存在，并且分布在左右两边(1 - 2, 3 - 4, 5 - 6, ... 在一边，2 - 3, 4 - 5, 6 - 7, ... 在另一边)，每一边的线两两之间必须包含/相离，不能部分相交(现实里就是纸条自己穿过自己了)。利用类似合法括号序列的方式可以单组 $O(n)$ 检查，本题没有卡暴力。

K. Haruhi with K-intimacy

日常没人交的题。

看到字符串前缀应该不难想到 *trie*。

考虑怎么在 *trie* 上统计答案。记 cnt_p 为经过点 p 的串的数量， ed_p 为以 p 结尾的串的数量。设 p 的深度为 dep ，设 k 是 p 的某个儿子，显然它对 $f(dep)$ 的贡献就是 $cnt_k \times (cnt_p - cnt_k)$ 。然后题目让我们求的是有序对，所以对每个儿子累加贡献并不会重复计数。唯一漏掉的是完全相同的字符串们产生的贡献，很好算就是 $ed_p \times ed_p$ 。

总结一下，我们只需要遍历整棵 *trie*，经过深度为 dep 的点 p 的时候统计

$$f(dep) = f(dep) + \sum_{k \text{ is a son of } p} cnt_k \times (cnt_p - cnt_k) + ed_p \times ed_p$$

就行了。

然后求一下 $f(i)$ 的前缀和，对于每个询问就可以 $O(1)$ 回答啦。

L. Haruhi with Literature club

二分贪心，每一个机器人的行动轨迹可以贪心定为优先把自己左侧没扫过的地扫一遍，如果还有 ≥ 2 的步数，则扫右边的。答案上限在 $2 * n$ ，如果使用 $mid = (l + r) / 2$ 二分， $l + r$ 可能爆 int 。

M. Haruhi with Mogician

First solved: 邓浩然, 00:01 (+).

邓总手真快啊。

C 语言课后习题，不用我多说了8。

