

CodeForces Round 532 (Div.2) 解题报告

SGColin

目录

1	A. Roman and Browser	2
1.1	Description	2
1.2	Solution	2
2	B. Build a Contest	2
2.1	Description	2
2.2	Solution	2
3	C. NN and the Optical Illusion	3
3.1	Description	3
3.2	Solution	3
4	D. Dasha and Chess	3
4.1	Description	3
4.2	Solution	3
5	E. Andrew and Taxi	4
5.1	Description	4
5.2	Solution	4
6	F. Ivan and Burgers	4
6.1	Description	4
6.2	Solution	4

1 A. Roman and Browser

1.1 Description

CodeForces 1100 A

给出一个长度为 n 的 01 序列 a ，以及一个 k 。你要找出一个 b ，使得所有 $c = b + ik \in [1, n]$ 的位置 a_c 删除后，序列内剩余的 01 个数差的绝对值最大。

1.2 Solution

枚举 b ，把所有对应位置删除后统计答案即可。

2 B. Build a Contest

2.1 Description

CodeForces 1100 B

给出一个值 n 和一个长度为 m 的序列 s 。

维护一个 multiset，扫描 s ，每次把 s_i 加入到 multiset 里，如果当前集合中 $1 \dots n$ 都各出现了一次，输出 1，并将集合中的 $1 \dots n$ 各去除一个，否则输出 0。

2.2 Solution

智障做法：维护值域线段树，节点记录当前代表值域有多少个位置不为 0，每次合法后将线段树重构，重构次数是 $O(\frac{m}{n})$ 的，所以复杂度 $O(m \log n + 2n \times \frac{m}{n}) = O(m \log n)$ 。

神犇做法 1：维护值域线段树，节点记录代表值域出现次数最小值，合法后打全局 -1 标记。

神犇做法 2：维护不同元素的计数数组，把他们都放到一个 multiset 里，每次更新就把老的计数数组 erase 掉，然后更新后再插进去。初始化 $ans = 0$ ，如果当前最小值是 ans ，则输出 1，令 $ans = ans + 1$ 即可。

神犇做法 3：对计数数组 cnt 开桶 bkt，记录当前合法次数 ans 。如果当前 $bkt[ans] = 0$ 证明所有的数字出现次数都超过了 ans ，输出 1，令 $ans = ans + 1$ 即可。

```
bkt[0] = n = rd(); m = rd();
for (rg int i = 1, x; i <= m; ++i) {
    x = rd();
    --bkt[cnt[x]]; ++bkt[++cnt[x]];
    if(!bkt[ans]) { putchar('1'); ++ans; }
    else putchar('0');
}
```

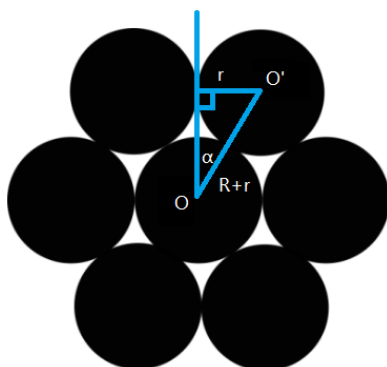
3 C. NN and the Optical Illusion

3.1 Description

CodeForces 1100 C

给出一个大圆的半径 R ，求正好外切 n 个互相相切的等大的圆，这些圆的半径 r 是多少。

3.2 Solution



$$\alpha = \frac{2\pi}{n} \times \frac{1}{2} = \frac{\pi}{n}$$
$$\frac{r}{R+r} = \sin(\alpha) \Rightarrow r = \frac{R \sin(\alpha)}{1 - \sin(\alpha)}$$

4 D. Dasha and Chess

4.1 Description

CodeForces 1100 D

交互题。有一个 999×999 的棋盘，初始你执一个白棋，对手有 666 个黑棋。白棋可以移动到边界以内的非空的八连通格子里，黑棋每次只能移动一个，但可以移动到任意一个空的格子里，你先手。规则是 2000 步以内只要有一个黑棋与你同行或同一列就认为你赢。

4.2 Solution

策略是先走到棋盘正中间 $(\frac{1+999}{2}, \frac{1+999}{2}) = (500, 500)$ 的位置，这个过程最多花费 1000 步的代价。此时将棋盘划分成四个象限，拥有黑棋数最少的象限的黑棋数不会超过 $\lfloor \frac{666}{4} \rfloor = 166$ 个，所以剩下的三个象限里黑棋总数不少于 500 个。此时直接走，与最少象限相反的对角线即可胜利。

能够赢得原因是，你走到边界最多只需要移动 499 步，在这 499 步中你的棋子每次卡住新的一行或一列，而对手没有足够的至少 500 步把剩下三个象限里的棋子移走。

值得注意的是，移动到 $(500, 500)$ 的过程中我们每次只动 x, y 坐标中的一个。不会出问题原因是如果将要移动到的格子有棋子，那么之前它和白棋就在同一行或同一列，此时对方已经输了。而后一步我们为了满足 500 步之内到达边界的限制必须走对角线，此时将要移动到的格子里可能有黑棋，需要特判。这一步我们只要改 x, y 坐标中的一个即可，不需要担心后续的问题，因为这一步之后白棋就与某一个黑棋处于同一行或同一列，对手就输了。

5 E. Andrew and Taxi

5.1 Description

CodeForces 1100 E

给出一个有向图，边有边权，现在你可以反向若干条边，求把图变成 DAG 所需要反向的边的边权最大值最小为多少。要求输出该条件下的操作方案。

5.2 Solution

最大值最小二分答案，转化为只能将边权小于二分值的边反向，是否能够造出 DAG。

巧妙的做法。先忽略所有边权小于二分值的边，看剩下的图是否是一张 DAG，如果是就有解。首先必要性显然。证明充分性的构造方法是，在拓扑排序判断是否为 DAG 的过程中记录拓扑序，剩下的边中如果有后被拓扑到的点指向先被拓扑到的点的边就反向。这样所有边都满足由拓扑序小的点指向拓扑序大的点，显然这样不会有环。

值得注意的是最后需要重新拓扑一遍，因为最后一次二分的值可能不合法。

6 F. Ivan and Burgers

6.1 Description

CodeForces 1100 F

设 $x = 2^{2^{100}} - 1$ 。给出一个序列，每次询问一个区间，求这个区间的一个子集异或和 y ，使得最大化 $x - x \oplus y$ 。

6.2 Solution

发现对于给定数据范围 $2^{2^{100}} - 1$ 就相当于无穷个 1，所以问题是区间内选择一个子集，使得异或和最大。

基础的想法是线段树，每个节点维护区间里的线性基，然后查询直接线性基合并。

有一个比较高妙的离线做法，把询问按照右端点排序，然后扫描到一个位置就回答右端点为当前位置的所有询问。此时维护的线性基是一个前缀线性基，那么如何体现左端点的限制呢？只需在线性基每个位置记录当前元素在序列中的位置，然后取答案的时候忽略掉不合法的位置即可。

此时还要保证每一位尽可能地新。做法是插入的时候如果当前位有值，先比较在原序列中的位置，保留靠后的一个，然后交换当前元素和原来该位置的元素，继续向下扫描。这样就能保证越高的位值越大。关于插入的实现细节，需要注意交换的时候换下来的值需要异或上当前该位上的值，并且注意判断是否需要比较在原序列中的位置，因为换下来的元素不一定在序列很靠后的位置。具体实现见 代码。

这样复杂度就是 $O((n+q) \log k)$ 的了，其实注意到线性基并没有多大，所以我们可以保存每一个前缀版本的线性基，此时询问就可以在线做了。