

A	B	C	D	E
+	+	+	+	+

rank:38

(竟然AK了，虽然是VP)

A

- 题意:找出一个最长的连续的子序列使得它不被 x 整除 ($1 \leq x \leq 10^4$)
- 题解:想不出来，暴力上了个线段树维护每个余数对应的最长前缀和。其实可以证明答案一定是前缀或者后缀，所以只用扫一遍就可以了。

B

- 题意:给定一个序列，从中选出一个最短的子序列，使得 $|s_1-s_2|+|s_2-s_3|+\dots+|s_{k-1}-s_k|$ 最大。
- 题解:找到所有拐点，删掉中间多余的点即可。

C

- 题意:给出序列 a_i 保证 $a_i \leq a_{i+1}$ 要求构造一个序列 b_i 使得 $\text{MEX}(a_1, a_2, \dots, a_i) = b_i$ 要求 $0 \leq b_i \leq 10^6$ 或判断无解。
- 题解:无解的情况即 $b_i \geq i$ 判断一下即可。有解的话，如果 a_i 在递增，那就令 $b_i = a_i$ 即可。否则如果 a_i 保持不变，那么可以将这些下标先入栈，然后一旦 a_i 发生变化，就从中栈中弹出元素弥补空缺位置，然后让 $b_i = a_{i-1}$ 即可。

D

- 题意:给出一个无向连通图。要么找到一个点数 $\leq k$ 的环，要么找到一个恰好有 $\lceil \frac{k}{2} \rceil$ 个点的独立集。
- 题解:之前还是这个出题人出过一个“Ehab大定理” Ehab's Last Theorem 那题就是把 k 和 $\lceil \frac{k}{2} \rceil$ 都改成了 \sqrt{n} 解法几乎一致，就是先dfs生成树找环，然后再找独立集。那题找独立集还需要注意一下细节，这题想都不用想直接暴力找就可以了，写的时候并没有想明白为什么还是可以ac 证明如下：如果所有环都大于 k 那么我们一定可以找到一个大小是 k 的子图，且这个子图是一颗树，一棵 k 个点的树一定可以找到一个恰好有 $\lceil \frac{k}{2} \rceil$ 个点的独立集。

E

- 题意:交互题，猜一个 0 到 $n-1$ 的排列。每次可以询问两个不同位置值的OR 最多询问 $4269(4396)$ 次 ($3 \leq n \leq 2048$)

- 题解:我们只要获得\$0\$的位置就可以通过\$n-1\$次询问获得剩下每个位置的数,因此我们要在\$2^{22}\$次获取到\$0\$的位置。首先我们有两个比较显然的引理:1.如果\$a|b > a|c\$那么\$b \neq 0\$。2.如果\$b \neq c, a|b = a|c\$那么\$a \neq 0\$。因此我们可以从第一个位置开始,假设某个位置对应的值\$a\$是\$0\$,然后依次询问它和后面每个值的\$OR\$同时维护\$a|b\$最小值所在的位置。如果没有出现相等的值,那么我们每次可以排除一个不是\$0\$的值;如果出现相等的值,那么原本那个位置就不是\$0\$,只有可能\$a|b\$最小值所在的位置是最小值,因此以这个位置为\$a\$继续往下扫即可。然而这里其实有一个问题,就是如果这个最小值是重复的值,那么有\$2\$个位置都有可能,因此我们要选一个位置然后额外询问一次这两个位置,这会造成一次额外询问,因此如果不随机访问是可能会被卡的。(然而我忘记随机访问但还是a子,可能数据就是随机的,当然这也证明了随机的情况下额外询问的次数是很小的)最后我们会剩下一个假设是\$0\$的位置以及后面位置中\$a|b\$最小值所在的位置,只有这两个位置可能是\$0\$,那么它们两个的\$OR\$值就是其中不为\$0\$的那个值,设为\$x\$。我们先假设其中一个位置的值是\$x\$然后用剩下的次数去随机访问其它位置和这个位置的\$OR\$。如果返回的答案\$y\$不完全包含\$x\$的所有二进制位,那么说明这个位置是\$0\$,否则这个位置就是\$x\$而另一个位置是\$0\$。事实证明剩下的次数还是比较多的,可以让这种做法的错误率降到很低。

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:farmer_john;jileo:codeforces_round_649_div_2_virtual_participation

Last update: 2020/06/19 23:40

