



(/wiki/index.php?title=Main_Page)

Log in (/wiki/index.php?title=Special:UserLogin&returnto=ACM-ICPC+2018+Xuzhou+Online+Contest)

Search EOJ Wiki



Page (/wiki/index.php?title=ACM-ICPC_2018_Xuzhou_Online_Contest)

Discussion (/wiki/index.php?title=Talk:ACM-ICPC_2018_Xuzhou_Online_Contest&action=edit&redlink=1)

View source (/wiki/index.php?title=ACM-ICPC_2018_Xuzhou_Online_Contest&action=edit)

History (/wiki/index.php?title=ACM-ICPC_2018_Xuzhou_Online_Contest&action=history)

ACM-ICPC 2018 Xuzhou Online Contest

Contents

- 1 ECNU Foreigners
 - 1.1 Problem A
 - 1.2 Problem B
 - 1.3 Problem C
 - 1.4 Problem D
 - 1.5 Problem E
 - 1.6 Problem F
 - 1.7 Problem G
 - 1.8 Problem H
 - 1.9 Problem I
 - 1.10 Problem J
 - 1.11 Problem K
- 2 One,Two,Three,AK
 - 2.1 Problem A
 - 2.2 Problem B
 - 2.3 Problem C
 - 2.4 Problem D
 - 2.5 Problem E
 - 2.6 Problem F
 - 2.7 Problem G
 - 2.8 Problem H
 - 2.9 Problem I
 - 2.10 Problem J
 - 2.11 Problem K

ECNU Foreigners

代码 (<https://github.com/F0RE1GNERS/JisuankeEatsMyCode/tree/master/1557>)

Problem A

Solved by ultmaster. 01:19 (+)

题意：一个环形的 n 的列表每个数都是 0 到 $2^k - 1$ ，求相邻两个数同或不为 0 的方案数。

题解：没看到环形的自闭了好久。Clarification 无响应。AC 了之后发了个样例解释在 Clarification 里居然有人说是错的（后来仔细一看 真的错了）。

观察一下或者打个表可以发现，异或和同或基本是一样的。除了 n 为偶数的时候，同或多 2^k 种方案。

这样就转化成异或的问题，这是一个经典的容斥问题。答案就是 $\sum_{i=0}^{n-1} \binom{n}{i} 2^{k(n-i)} (-1)^i$ 。

本来想用 OEIS 的（表都打好了，又读错了题有点自闭），结果一下子就推出来了。。。

Problem B

Solved by kblack. 01:16 (+1)

题意：博弈，每人轮流选择可以 $+a_i$ 或 $-b_i$ 或 $\times(-1)$ ，两个人分别要最终大于或小于某个数，求结果。

题解：记忆化搜索，状态是 (当前轮次, 当前分数)。

Problem C

Solved by ultmaster. 02:56 (+2)

题意：有点复杂。一个 1~9 的九宫格，有些东西别人知道你不知道，有些东西大家都不知道。别人会采取最优策略选横着的或者竖着的或者斜着的三个数，然后获取一定的积分（因为有的数 TA 也不知道，所以是一种策略）。你要计算别人得分的期望。

题解：对于你不知道的别人知道的每一种情况，要算出得分期望的最大值。得分期望的最大值是每种决策（总共 8 种）得分期望中最大的一种。想清楚了这一点模拟就好。实现的时候由于常数巨大（甚至多了个 log），改成 pb_ds hash_table 才过。

Problem D

Solved by zerol. 00:59 (+2)

题意：求 $\sum_{i=1}^m \mu(in)$

题解：如果 n 中有平方因子，那么显然答案是 0。否则相当于求 $\mu(n) \cdot \sum_{i=1}^m \mu'(i)$ ，其中 μ' 是在 μ 的基础上不把 n 的质因数当质数。类似于求 μ 的前缀和的方法（任意一种亚线性筛），改一改就能过了。

min_25 筛的做法：对于求 μ ，第一步是求质数的个数的前缀和，先假装所有质数都是质数，然后对于 n 的质因子，把它们从质数个数前缀和中除名。第二步计算前缀和的时候就按照真实 ($f(p^c) = -1$ 当且仅当 $c = 1$ 且 p 不是 n 的质因数) 的方法计算。

杜教筛的做法：

$$\begin{aligned}
 f(n, m) &= \sum_{i=1}^n \mu(im) \\
 &= \mu(m) \sum_{i=1}^n [(i, m) = 1] \mu(i) \\
 &= \mu(m) \sum_{d|m} \mu(d) \sum_{i=1}^{\lfloor \frac{n}{d} \rfloor} \mu(id) \\
 &= \mu(m) \sum_{d|m} \mu(d) f(\lfloor \frac{n}{d} \rfloor, d)
 \end{aligned}$$

递归部分不用记忆化，暴力递归，复杂度玄学。递归的中止条件是 $b = 1$ 此时就变成求 μ 的前缀和了，可以用杜教筛在 $O(n^{\frac{2}{3}})$ 时间内求出所有需要的 $f(a, 1)$ （可以参考 2016 集训队论文）。

zerol: min_25 筛的好处是不用思考，但这题好像魔改的有点多（所以玩脱了）。如果 min_25 第一步直接计算 ($f(p)$ 的函数值只有 0 或 1，满足完全积性的条件) 的话会需要 $O(\sqrt{m})$ 次 1~x 中与 n 互质的数的个数，这部分会算得很慢，所以才要先把所有质数当质数在扣除掉 n 的质因子。

Problem E

Upsolved by zerol.

题意：求至多经过 k 个点的点权和最大的路径的点权，其中连续的 t 个点的点权可以翻倍。

题解：如果没有翻倍操作，那么直接快速幂可以求出答案（直接矩阵快速幂）。现在先求不超过 t 个点的答案矩阵 R ，将其翻倍。然后需要在前面或后面补上共 $k-t$ 条边。最朴素的做法是在 R 前后各乘一个矩阵然后取较大值（含义就是要么前面加一条边，要么后面加一条边），重复 $k-t$ 次。现在要用快速幂加速这个过程，我要保证 $k-t$ 的任意拆分 ($a+b=k-t$) 能够被表示。所以对 $k-t$ 进行拆分，先拆出 1,2,4,8,... 剩下的部分再补上去（比如 20 会被拆成 1,2,4,8,1,4）。

更详细的题解： $A[i][i] = v[i]$, $G[i][j] = v[j]$, $rG[i][j] = v[i]$ ($\forall (i, j) \in E$)。先求出至多 t 个点的答案， $R = A \times G^{t-1}$, $f(M) = rA \times M + M \times A$ ，那么答案就是 $f^{k-t}(2R)$ 。定义乘法就是矩阵乘法中的乘改成加，加改成 max，然后对两个矩阵都取 max。定义加法就是 max。

zerol: 经过讨论一致决定，这锅卡车运输背。

Problem F

Solved by ultmaster. 00:26 (+)

题意：每天出现若干种颜色，求最长的时间区间使得一种颜色连续出现。

题解：维护每种颜色已经连续出现的次数即可。

Problem G

Solved by kblack. 00:34 (+)

题意：堆叠若干个以原点为左下角的矩形，求看得到的右边界和上边界的总长度。

题解：从后往前，添加矩形时计算向下和向左露出的边界长度，这个部分可以用 bit 或线段树维护后缀最大值做。

Problem H

Solved by zerol. 01:16 (+)

题解：单点修改，询问区间内所有前缀和的和。

题解：用 BIT 维护 $a[i]$ 以及 $(n-i+1)*a[i]$ 的区间和，一次询问就是 $(n-i+1)*a[i]$ 的区间和减去若干倍的 $a[i]$ 的区间和。

Problem I

Solved by kblack. 00:16 (+)

温暖的签到 A。

Problem J

Solved by kblack. 02:42 (+3)

题意：造一个最便宜的迷宫，求一个最短路径。

题解：造一个最贵的生成树，求一个树上距离。

Problem K

Solved by zerol. 02:51 (+6)

题意：求模 2 意义下很多次二维卷积的结果。

题解：类似快速幂，但要求出一次卷积的效果（每一个位置由所有位置中若干个位置 1 的个数的奇偶性唯一决定），总复杂度 $O(n^4 \log t)$ 。

更加详细的题解：

需要解决的问题有：计算一次卷积的效果，把一个效果翻倍，把一个效果作用于输入的矩阵。

一次卷积的效果：对于矩阵 A 的每一个位置，用一个二进制位标记。最后的效果表示为每个位置附带有一个二进制数，表示所有位置对该位置产生贡献是奇数次还是偶数次。

将效果作用于矩阵：对于每个位置，计算出其他所有位置对它的贡献之和奇偶性。

效果翻倍：对于每个位置，计算所有位置对它贡献之后的所有位置的新的奇偶性。

计算一次卷积的效果：跑一趟卷积就好了。

One, Two, Three, AK

Problem A

Solved by Xiejiadong. 2:15:12(+)

题意：一个环形的 n 的列表每个数都是 0 到 $2k-1$ ，求相邻两个数同或不为 0 的方案数。

题解：一开始没开这题。写了几个数以后，一下就推出来了，似乎并不难。

对于 n 分奇数和偶数讨论一波就好了。

Problem B

Solved by oxx1108. 1:38:59(+1)

题意：两人博弈，每个人有三种选择，最后问谁必胜或者平局。

题解：套路题，记忆化搜索一下就行。

Problem C

Solved by oxx1108. 3:15:01(+1)

题意：算个填九宫格的期望。

题解：直接模拟一下就好了，时间复杂度极限9!，就是写起来比较麻烦。

Problem D

Solved by dreamcloud. 4:48:53(+3)

题意：求 $\sum_{i=1}^m \mu(in)$

题解：

$$\begin{aligned}
 f(n, m) &= \sum_{i=1}^m \mu(in) \\
 &= \mu(n) \sum_{i=1, \gcd(i, n)=1}^m \mu(i) \\
 &= \mu(n) \left(\sum_{d|n} \mu(d) \sum_{i=1}^{\lfloor \frac{m}{d} \rfloor} \mu(id) \right) \\
 &= \mu(n) \left(\sum_{d|n} \mu(d) f(d, \lfloor \frac{m}{d} \rfloor) \right)
 \end{aligned}$$

Problem E

Unsolved.

Problem F

Solved by oxx1108. 1:00:47(+4)

题意：求某个坐标最长连续出现的次数。

题解：直接map哈希，然后扫一下就行。忘记clear wa了四发。

Problem G

Solved by Xiejiadong. 4:43:03 (+6)

题意：每一次的波浪就会将自己范围内的痕迹清除，求所有波浪后的痕迹长度。

题解：我们对于所有的波浪倒着处理。用当前区间的最大值和自己的高度处理一下就行了。用线段树来维护。

煞笔题...从开场卡到结束。

Xiejiaotong：我又来背锅了。

Problem H

Solved by dreamcloud. 1:03:30(+)

题意：给你个数组 $a[1] \dots a[n]$ ，两个操作 1.查询 $[l,r]$ ，输出 $\sum_{i=l}^r a[i] * (r - l + 1)$ 2.更新 $a[pos]$

题解：用树状数组维护两个前缀和，一个是 $a[i]$ 的前缀和，一个是 $a[i] * (n - i + 1)$ 的前缀和。

Problem I

Solved by oxx1108. 0:22:20(+1)

题意：签到

题解：签到

Problem J

Solved by Xiejiaotong. 02:42 (+1)

题意：在图中造一些墙，使得图中任意两点之间的路径在最小花费的情况下唯一，给出 q 个询问，每次询问任意两点之间的最短距离。

题解：每一格都向下方和右方连出一条边，在这些边中，找出一些边，使得路径唯一，并且花费最小。

这显然就是最大生成树，在最大生成树上，跑一下LCA就能计算最短距离了。

Problem K

Unsolved.

This page was last edited on 11 September 2018, at 18:34.



(//www.mediawiki.org/)