牛客暑期ACM多校训练营

第4场-zimpha

牛客网
NOWCODER

A. Ternary String

- 给个长度为n的三进制串，有这样一个操作：在每个2后面插入一个1，每个1后面插入一个0，然后删掉第一个字符。问多少次操作后，变成空串
- n <= 1e5

A. Ternary String

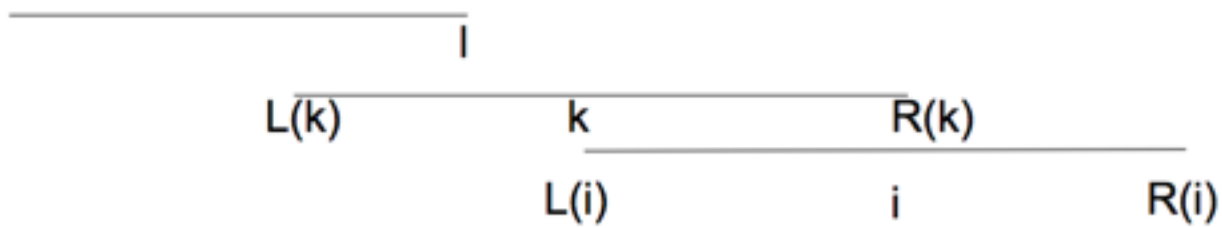
- 操作分析
  - 遇到0，直接删除，操作次数+1
  - 遇到1，考虑之前已经操作了x次，那么这个1后面已经多生出了x个0。这个时候需要再经过x+2次操作才能删完
  - 遇到2，考虑之前已经操作了x次，然后打个表可以发现，之后还需要经过3 * (2 ^ (x + 1) - 1) - x次操作才能全部删完
- 可见需要计算a*2^x mod (10^9 + 7)，于是也得计算x mod phi(10^9+7)，考虑到x可能也是之前某些2^x的组合，因此还得算x mod phi(phi(10^9+7))，依次类推，phi(phi(phi(10^9+7))), …, phi^k(10^9+7)都得计算。
- 另外考虑到若干次后，mod一定是一个2的倍数，可以提前算好答案，来优化。

B. Interval Revisited

- 给出n个带权区间，选择若干区间，覆盖[1, m]，使得每个位置被覆盖权值和的最大值最小
- $n, m <= 2000$

B. Interval Revisited

- 一个显然的结论：每个位置最多被两个区间覆盖
- 所有区间按照右端点从小到大排序
- $dp(i, j)$表示第i个区间必选，上一次被覆盖一次的位置是j，$[1, j]$覆盖权值和最大值的最小值
  - $dp(i, j) = \min(\max(dp(k, l), w(i) + w(k)))$ $R(k) + 1 >= L(i)$ and $L(k) - 1 <= l < L(i)$
  - j和$R(k)$有关，显然第二维就是个区间最值，线段树优化下



C. Chiaki Sequence Reloaded

- 求下面数列前n项和

$$a_n = \begin{cases} 0 & n = 1 \\ a_{\lfloor \frac{n}{2} \rfloor} + (-1)^{\frac{n(n+1)}{2}} & n \geq 2 \end{cases}$$

- n <= 1e18

C. Chiaki Sequence Reloaded

- 下面这个式子很容易从定义得出来

$$a(n) - a([n/2]) = \begin{cases} 1 \text{ if } n \equiv 0 \text{ or } n \equiv 3 \pmod 4; \\ -1 \text{ if } n \equiv 1 \text{ or } n \equiv 2 \pmod 4. \end{cases}$$

- 令n的二进制表示为b(m)b(m-1)…b(0)，且b(m)=1
  - p(n)是b(i)=b(i+1)的i个数，q(n)是b(i) != b(i+1)的i个数
  - 显然a(n)=p(n)-q(n)
- 之后考虑数位dp计算p(n)-q(n)
  - dp(i, j,x,y)表示前i位和为j，最后一位是x，和n大小关系是y的方案数。
  - 预处理一些东西，O(log^2 n)单询问，再优化下甚至可以O(log n)

D. Another Distinct Values

- 给出n，构造nxn的矩阵，每个元素是0，-1,1，使得每行每列的sum互不相同
- n <= 200

D. Another Distinct Values

- 总和范围是[-n, n]，共2n+1种，令r(1), r(2), .., r(n), c(1), c(2), …, c(n)是行列的和，显然xjb排列下可以得到r(1), r(2), …, r(k) >= 0, c(1), c(2), …, c(n-k) >= 0
- 考虑下面两个式子

$$\sum_{i=1}^{n} |r_i| + \sum_{j=1}^{n} |c_j| \ge \sum_{r=-n}^{n} |r| - n = n^2.$$

$$\begin{aligned}
\sum_{i=1}^{n} |r_i| + \sum_{j=1}^{n} |c_j| &= \sum_{i=1}^{k} r_i - \sum_{i=k+1}^{n} r_i + \sum_{j=1}^{n-k} c_j - \sum_{j=n-k+1}^{n} c_j = \\
&= \sum_{i\le k} a_{ij} - \sum_{i>k} a_{ij} + \sum_{j\le n-k} a_{ij} - \sum_{j>n-k} a_{ij} = \\
&= 2\sum_{i=1}^{k}\sum_{j=1}^{n-k} a_{ij} - 2\sum_{i=k+1}^{n}\sum_{j=n-k+1}^{n} a_{ij} \le 4k(n-k).
\end{aligned}$$

D. Another Distinct Values

- n^2 <= 4k(n-k)，(n-2k)^2 <= 0，于是n=2k，是个偶数
- 考虑如下构造法，从2x2出发，依次扩展2层

1	0
1	-1

<table>
<tr><td colspan="4" rowspan="4">$n \times n$</td><td>1</td><td>1</td></tr>
<tr><td>-1</td><td>-1</td></tr>
<tr><td>1</td><td>1</td></tr>
<tr><td>-1</td><td>-1</td></tr>
<tr><td>1</td><td>-1</td><td>1</td><td>-1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>-1</td><td>1</td><td>-1</td><td>1</td><td>-1</td></tr>
</table>

E. Skyline

- 平面上n个点，每个点出现概率是p_i，求期望被支配的点数
- 一个点被支配，当且仅当存在一个点在其右上方
- n <= 1e5

E. Skyline

- 同一位置上的点可以先把概率合并掉，$p = 1 - \mathrm{prod}\,(1-p_i)$
- 枚举点i和j，他们一定出现，考虑贡献
    - $x(i) < x(j)$，$y(i) > y(j)$，支配点数是$(x(j)-x(i)) * y(j)$
    - 概率是$p(i) * p(j) *$ 一堆其他数
        - $x= x(i), y > y(i)$的
        - $x >= x(j), y > y(j)$的
        - $x(i) < x < x(j), y > y(j)$的
- 考虑分治，这些东西都是可以用个线段树维护下，然后用个区间求和计算

F. Beautiful Garden

- n x m的格子，选个p x q的，中心重合，且上下左右对称
- 求(p, q)方案数
- n, m <= 2000, n, m是偶数

F. Beautiful Garden

- 直接模拟即可

G. Maximum Mode

- n个数，删m个数，使得剩下数里面众数最大
- m < n <= 1e5

G. Maximum Mode

- 枚举答案，考虑需要删几个数才能使之变成众数
    - 出现次数多余它的都要被删到次数比它小
    - 剩下的随便删

H. Double Palindrome

- 长度为n的字符串，可以交换两个位置，使得最后是双回文，求交换方案
- $n <= 1e5$

H. Double Palindrome

- 枚举i，要求s[1..i]和s[i+1..n]都是回文串
  - 两边都不是回文串：需要从左右各选个位置交换
  - 两边都是回文串：只要是相同的位置都可以交换，用个flag记住
  - 其中一个是回文串，另一个不是：另一个选两个位置交换
  - 还需要考虑奇回文串的中心也可以用来交换
- 除去相同元素的交换，其他交换次数是O(n)的，于是暴力拿set去重
- 大概得写个manacher和后缀数组

I. Permutation

- 给出n对三元组(a[i], b[i], c[i])，保证3n个数互不相同且在1到3n之内。构造一个排列使得b[i], c[i]出现在a[i]后面，并且sum |p[i] - p[i-1]|最小
- n <= 1e5

I. Permutation

- 考虑抽象成这样一个模型：
  - 数轴上[1,3n]内的整点，你要走遍这3n个点，一个点可以经过多次
  - 限制：存在a[i]出现在b[i]和c[i]前面
  - 最短路
- 显然上述模型和原问题等价，知道上述一条路径，可以直接从路径中还原排列
- 一个显然的结论：每个位置最多经过3次
- 推论：一定是下面两种走法，直接dp走法即可

J. Hash Function

- 给出一个基于linear probing的hash table，求一个字典序最小的插入序列，或者判断不合法
- n <= 2e5

J. Hash Function

- 从hash表里能得出的信息：
    - 一个区间里面的数要比一个数插入的早
- 如果一定有解
    - 根据这些信息可以建出一个图，然后拓扑排序可以得到答案
    - 线段树优化建图：边数O(n^2) => O(n)
- 如何判不合法：
    - x最终插入到j，应该要插到i=x mod n，如果[i..j-1]里面有-1，
    - 拓扑排序出现环

Thanks

牛客网