CF水题选讲

绍兴一中 季雨田

578E. Walking!

No.13

578E. Walking!

- 【题目大意】有n个排成一排的点，你可以从任意一个点出发，左右脚交替地不重不漏地踩过所有点。一开始的左右脚也由你自己决定。给出每个点被踩到的左右脚，请你构造一种满足条件的方案，且往左走的步数最少。输出任意一个合法方案即可，保证有解。
- 【数据范围】$n \le 10^5$

578E. Walking!

- 首先可以发现左脚（L）和右脚（R）的数量之差肯定不超过1。而且，如果数量不等，第一步是L还是R就已经确定了。
- 然后可以发现，对于一串LR的脚印，我们可以把它分成若干段LR交替的子序列，设最小的划分数量为k，那么最小的回头数肯定不小于k。
- LR交替的子序列有4种，分别是L开头L结尾（$LR \dots RL$），L开头R结尾（$LR \dots LR$），R开头L结尾（$RL \dots RL$），R开头R结尾（$RL \dots LR$）。
- 如果我们可以改变这4种子序列的顺序，使得它们拼接起来还是LR交替的，那么我们就能完成这题了。

578E. Walking!

- 当得到的k个序列中至少存在一个LL或RR时，做法就很简单了。如果LL比RR多1，那么把它们按如下方法进行拼接：$LLRR \dots RRLL$。得到一个LL，然后再把LR都拼在LL的左边，RL都拼在LL的右边。RR比LL多1同理。
- 如果LL和RR一样多，用之前的拼法使得LL和RR的数量恰好为1，若存在RL则这么拼$(LR)LLRLRR$，若存在LR同理$RRLRLL(RL)$。
- 如果不存在LL或RR，那么随便找一对LR和RL，假设LR的最后一个R比RL的最后一个L位置更靠后，那么直接把LR的R接到RL的后面即可得到一个LL和一个RR。反之同理。

578E. Walking!

- 那么最后的问题就是怎么求出最小的k和这k个序列。
- 贪心，每次找最近的L或R，直到找不到再开始新的一轮，直接用并查集或其他数据结构维护即可。
- 时间复杂度$O(n\log n)$，空间复杂度$O(n)$。

788D. Recover a functional graph

No.12

788D. Recover a functional graph

- 【题目大意】交互题，平面上有一些平行于x轴和平行于y轴的直线。每次询问你可以给出平面上的一个点，交互库会给出与这个点距离最近的直线的距离。你需要确定所有的直线。
- 【数据范围】平行于x轴和平行于y轴的直线各不超过10^4条，坐标范围的绝对值不超过10^8。询问次数不能超过3×10^5。

788D. Recover a functional graph

- 首先一个想法是询问形如(x,x)的点，当得到的回答为0时，则问一下点(x,t)和点(t,x)以确定是否存在与x轴和y轴平行的直线，其中要保证(t,t)的回答不是0 。
- 于是倍增即可，每次x加上询问(x,x)的回答。
- 需要注意的是这题稍微卡了一点常数，发现倍增的主要步数浪费在了前几步上，于是从0的回答出发时先试着问一下$(x+10^3,x+10^3)$的回答是不是10^3，是的话直接跳，不是的话再从$x+1$开始问。

713E. Sonya Partymaker

No.11

713E. Sonya Partymaker

- 【题目大意】有n个人，m个凳子，其中凳子被编号为$1 \sim m$围成环形。1和m相邻，i和$i+1$相邻。第i个人在编号为a_i的椅子上。现在每个人可以选择顺时针或逆时针地走动x步，边走边把当前位置凳子标记，问将凳子都标记完的最小x。
- 【数据范围】$m \le 10^9$，$n \le 10^5$，$1 \le a_i < a_{i+1} \le m$

713E. Sonya Partymaker

- 显然可以二分答案，问题就转化成了是否能在每个人的步数都不超过x的情况下标记所有椅子。
- 注意到每个人走的步数都是一样的。
- 可以发现如果第i个人还没有走时，a_i前面有没有标记过的椅子，那么这个未标记的椅子肯定会被第i个人或第$i+1$个人标记。
- 因为如果被第$i+1$个人之后的人标记（设这个人是第j个），那么显然第i个人能标记a_i到a_j之间的所有椅子，于是让i往后，$i+1$往前，j往后，答案肯定不会变劣。

713E. Sonya Partymaker

- 设f_i表示处理了前i个人，最远延伸到的位置与a_i的差。
- 根据之前的讨论，我们一要考虑$f_i \to f_{i+1}$和$f_i \to f_{i+2}$即可，这两种转移要分别讨论$i+1$和$i+2$的方向。
- 如果直接从1开始DP可能会受到n的影响，于是可以找到一个间距大于$x+1$的两人（如果没有肯定合法）。然后以后面那个人为起点，因为起点之前必然存在椅子没有标记，所以讨论一下是起点往前，还是起点+1往前，分别DP即可。
- 总的复杂度是$O(n\log n)$。

744D. Hongcow Draws a Circle

No.10

744D. Hongcow Draws a Circle

- 【题目大意】平面上有n个红点，m个蓝点，你需要画一个最大的圆，满足圆内至少有一个红点，且没有蓝点，边界上的点可以视为在圆内，也可以不在。输出最大的圆的半径。无限大则输出-1。
- 【数据范围】$n, m \le 10^3$

744D. Hongcow Draws a Circle

- 首先考虑无限大的情况，如果红点不在蓝点形成的凸包内，那么显然输出-1。
- 如果答案有限，那么圆肯定与蓝点相切。
- 于是枚举蓝点，二分圆的半径，将圆绕蓝点旋转，可得到其他点相交的红色或蓝色圆弧。把圆弧投影到单位圆上，判断单位圆上是否存在一个点只被红弧包含。
- 这样复杂度就是$O(n^2\log^2 n)$。
- 如果随机枚举蓝点，期望只有 $O(\log n)$ 个蓝点需要二分，因为随机排列的前缀最值个数规模是$O(\log n)$。
- 于是复杂度就是$O(n^2\log n + n\log^3 n)$

618G. Combining Slimes

No.9

618G. Combining Slimes

- 【题目大意】有一个$1 \times n$的面板，两个值为x的数字块碰到一起会形成一个$x+1$的数字块。一开始面板上没有数字块。接下去会一直执行以下的操作，直到面板满了：在面板的最右端会生成一个数字块，有p的概率，这个数字块的值是1，有$1-p$的概率这个数字块的值是2。面板会向左倾斜，数字块会碰撞。求所有数字和的期望是多少，输出实数。
- 【数据范围】$n \le 10^9$，$0 < p < 1$

618G. Combining Slimes

- 首先可以发现一些很显然的性质：
- 最后的局面一定是由若干个单调递减的序列拼成的，且这些序列都以1结尾（最后一个序列可以以2结尾）。
- 一个$1 \times n$的面板最多只能产生权值为$n+1$的数字块。
- 当数字块的权值很大时，出现该数字块的概率小到可以忽略不计。
- 据估计，当数字块的权值大于50时，出现该数字块的概率小于10^{-300}。也就是我们只要考虑数字块的权值不超过50的情况。于是，问题就被大大化简了。

618G. Combining Slimes

- 设$a_{i,j}$表示在$1 \times i$的面板上操作，出现状态$(j, 0,0,0, \dots)$（即最左边的数字块是j，且没有其它数字块）的概率。
- 设$b_{i,j}$表示在$1 \times i$的面板上操作，出现状态$(j, 0,0,0, \dots)$且第一个生成的数字块是2的概率。
- 易得转移方程$a_{i,j} = a_{i,j-1} \times a_{i-1,j-1}$，$b_{i,j} = b_{i,j-1} \times a_{i-1,j-1}$。而且，由于我们只考虑$j$不超过50的情况，所以，当$i > 50$时，可以得到$a_{i,j} = a_{i-1,j}$，$b_{i,j} = b_{i-1,j}$。所以我们只要计算$a$和$b$的前50行和前50列即可。
- 假设第$i-1$格的权值比j大，那么我们可以利用$a_{i,j}$算出第i格恰好是数字块j的概率：$a_{n-i+1,j} \times \left(1 - a_{n-i,j}\right)$。

618G. Combining Slimes

- 设$f_{i,j}$表示第i格到第n格的期望权值和，其中第i格的权值为j，且这个数字块不会与其它数字块合并。
- 那么最后的答案就是:
- $ans=\sum_{j=1}^{50}a_{n,j}\times\left(1-a_{n-1,j}\right)\times f_{1,j}$。
- $f_{i,j}$的转移有两种:
- 当$j=1$时，$f_{i,j}=j+\frac{\sum_{k=2}^{50}f_{i+1,k}\times b_{n-i,k}\times\left(1-a_{n-i-1,k}\right)}{\sum_{k=2}^{50}b_{n-i,k}\times\left(1-a_{n-i-1,k}\right)}$;
- 当$j>1$时，$f_{i,j}=j+\frac{\sum_{k=1}^{i-1}f_{i+1,k}\times a_{n-i,k}\times\left(1-a_{n-i-1,k}\right)}{\sum_{k=1}^{i-1}a_{n-i,k}\times\left(1-a_{n-i-1,k}\right)}$。

618G. Combining Slimes

- 我们可以先算出$f_{N,j}$到$f_{N-50,j}$的值，因为当$i > 50$时，$a_{i,j} = a_{i-1,j}$，$b_{i,j} = b_{i-1,j}$，所以可以直接用矩乘算出$f_{1,j}$的值。
- 这样总的时间复杂度瓶颈在于矩乘：$O(50^3 \times \log n)$

794G. Replace All

No.8

794G. Replace All

- 【题目大意】对于两个仅包含大写字母A和B的字符串x和y，一个01串对(s,t)被称为不错的当且仅当：
  - $1 \le |s|,|t| \le n$
  - 且如果把x和y中所有的A替换成s，把所有的B替换成t，那么由x和y产生的这两个新字符串完全相同。
- 一个字符串对x和y的柔韧度被定义为对它来说不错的01串对(s,t)的对数。给你两个仅由A,B和?构成的字符串对c和d，其中所有的?都必须替换成A或者B。请你求出所有可能的字符串对的柔韧度之和，答案对10^9+7取模。
- 【数据范围】$|c|,|d|,n \le 3 \times 10^5$

794G. Replace All

- 先考虑没有?怎么做。
- 首先算出s和t的长度之比，例如$c = ABABAAA$，$d = BAABBB$，那么就必须有$3|s| = 2|t|$。
- 先考虑那些长度比值已知的情况。假设$|s|:|t| = x:y$（x, y互质），那么我们可以发现，当且仅当$s = w^x, t = w^y$时，(s,t)合法。其中w可以是一个任意长度的01串。
- 为什么？我们可以先将s表示成x个长度相同的块，将t表示成y个长度相同的块，然后将c和d产生的两个串对应的块连边，表示它们要完全相同。可以发现，除非c和d完全相同，否则s的每一个块都会和t中的每一个块连边。不理解的可以自己画一画。

794G. Replace All

- 然后我们就可以枚举w的长度计算一下，贡献是一个2的幂的前缀和的形式，统计到答案里。这样做一次的时间复杂度是$O(\log|c|)$（求gcd）的。
- 再来考虑长度之比未知，即c和d中AB个数恰好都相等的情况。
- 这种情况下，s和t的长度可以随意取，如果c和d不完全相同，那么每一块的长度就是$\gcd(|s|,|t|)$，算贡献可以枚举这个gcd，然后计算$1 \sim \left\lfloor \frac{n}{\gcd} \right\rfloor$中互质的对数，统计到答案里。
- 如果c和d完全相同的话，s和t可以任意取，一定合法。

794G. Replace All

- 现在考虑有?的情况，发现每个?具体是什么并不重要，我们只需要知道c和d中AB各有多少个。
- 于是枚举c,d中各有多少个?变成A即可，乘上一个组合数算一下，时间复杂度$O(|c|^2 \times \log|c|)$。
- 还要继续优化。
- 继续发现，不需要知道c和d中AB各有多少个，只需要知道它们的差就可以了。于是我们枚举c中的?变成A的个数减去d中的?变成A的个数，组合数算一下，时间复杂度为$O(|c| \times \log|c|)$，可以解决本题。

744E. Hongcow Masters the Cyclic Shift

No.7

744E. Hongcow Masters the Cyclic Shift

- 【题目大意】一个字符串集合X是合法的，当且仅当其满足以下条件。
  - 首先定义字符集合S_X为：一开始令$S_X = X$，之后如果$a \in S_X$且$b \in S_X$，那么把a和b的拼接加入S_X。易知S_X为无穷集合。
  - 认为一个字符串s是合法的当且仅当其满足以下条件：存在一种s的划分满足：s可以分成$w_1, w_2, \dots, w_k$，其中$w_i \in X$，且s的循环位移的$|s|$个串中恰好有k个串在S_X中。
  - 只有当S_X中的所有字符串都合法，X才被称为合法。
- 给出n个字符串，如果第l个到第r个字符串组成的集合合法，则$f(l,r) = 1$否则，$f(l,r) = 0$。求$\sum_{l \le r} f(l,r)$。
- 【数据范围】$n \le 30$，$\sum_{i=1}^{n} |s_i| \le 10^5$

744E. Hongcow Masters the Cyclic Shift

- 首先，为了方便描述复杂度，定义$m = \sum_{i=1}^{n} |s_i|$。
- 可以发现随着l的递增，合法的r单调不降，于是我们只要验证$O(n)$个合法的字符串集合。
- 将每个字符串的每个后缀建一个点，将每个后缀与所有字符串比配，若完全匹配，则将后缀与匹配之后剩下的后缀连边。
- 关于后缀和整串的比配，需要用到扩展kmp。
- 这样得到一张$O(m)$个点，$O(nm)$条边的有向图。

744E. Hongcow Masters the Cyclic Shift

- 如果这张图存在环，就说明不合法，因为存在环就说明存在一个循环串可以被表示成2种及以上的拼接。
- 这样子的话，可以直接dfs判环，时间复杂度与图中的边数有关，为$O(nm)$。
- 总的复杂度是$O(n^2m)$。

704E. Iron Man

No.6

704E. Iron Man

- 【题目大意】有一棵n个点的树，有m个人，每个人有四个参数(t, c, v, u)，表示这个人t时刻于点v出现，并以每秒c条边的速度向点u移动，到达点u后消失。问最早的相遇时刻。若$u = v$，则表示这个人仅在t时刻出现。
- 【数据范围】$n, m \le 10^5$，$t, c \le 10^4$

704E. Iron Man

- 首先考虑树退化成链怎么做，那么每个人都可以表示成在$t-x$坐标系上的一条线段。
- 于是问题变成了给出平面上的若干条线段，问交点的最小横坐标。
- 那么树上的话显然可以树剖。把x轴换成dfs序就行了。
- 把每个线段分成插入点和删除点，用set维护每个线段的高低关系，当删除一条线段时，如果之前比它高的线段现在比它低了，或之前低的变高了，那么就更新答案。
- 注意一下精度，这道题就解决了，复杂度$O(n\log^2 n)$

645G. Armistice Area Apportionment

No.5

645G. Armistice Area Apportionment

- 【题目大意】有两个点P和Q，坐标分别是$(a,0)$和$(-a,0)$，定义直线的权值为$max(|PX-QX|)$，其中X为直线上的任意一点。现给出平面上的n个点，问所有至少经过2个点的直线中权值最小的直线的权值是多少。
- 【数据范围】$n \le 10^5$，$|x|,|y|,a \le 10^4$

645G. Armistice Area Apportionment

- 考虑知道直线l怎么求权值，也就是$\max(|PX-QX|)$。
- 如果P和Q在直线l的同侧，那么显然$\max(|PX-QX|)=PQ$，同理，如果在异侧，那么设P'为点P关于直线l的镜像，于是可以得到$\max(|PX-QX|)=P'Q$。
- 接下来，我们考虑到直线l其实就是点P和P'的中垂线，而中垂线上的点到两端点的距离相等，也就是说$XP=XP'$。
- 同时，两点确定一条直线，于是我们以n个点为圆心，经过点P作n个圆。
- 那么圆与圆的交点就是P'所在的位置。

645G. Armistice Area Apportionment

- 接下来问题就转化成了，平面上有n个圆，求所有圆交点中与点Q最近的交点，输出该交点到点Q的距离。
- 这就是一个很简单的二分答案，求一个圆上的弧是否相交的问题了。
- 总的时间复杂度是$O(n\log^2 n)$。

750H. New Year and Snowy Grid

No.4

750H. New Year and Snowy Grid

- 【题目大意】有一张有障碍的$h \times w$的网格图，问是否存在一条路径，从左上角出发到右下角，再回到左上角，且不会重复经过同一个点（起点除外）。有q组询问，每组询问给出k个非障碍点，问是否存在不经过这k个点的合法路径。
- 【数据范围】$h, w \leq 10^3$，$q \leq 10^4$，$k \leq 10$

750H. New Year and Snowy Grid

- 首先，问题相当于求起点到终点是否存在两条不相交的路径。
- 如果只有一组询问的话，那就是一个经典的网络流模型，我们可以给每个点拆点，如果这个点是障碍点，那么i向i'连流量为0的边，否则，i向i'连流量为1的边，表示每个点只能经过一次。再每个点向其周围的4个点连流量为1的边，然后从S到T跑一边最大流，如果流量小于2，那么答案就是NO，否则就是YES。
- 接着我们可以把最大流问题转换成最小割问题，不过遗憾的是之前的建图并不是平面图，不能转换成最短路问题。

750H. New Year and Snowy Grid

- 我们可以用一个巧妙的办法，直接考虑最短路模型，S向所有$(i,1)(2 \le i \le n)$和$(n,j)(1 \le j < m)$连边，也就是最左边的那一列和最下边的那一行。所有$(i,m)(1 \le i < n)$和$(1,j)(2 \le j \le m)$向T连边，也就是最右边的那一列和最上边的那一行。
- 然后所有点向八联通方向连边，边没有边权，每个点有点权，障碍点的点权为0，空地的点权为1。
- 跑一遍最短路，如果最短路小于2，那么答案就是NO，否则就是YES。
- 这样的复杂度就是最短路的复杂度了，考虑到点权只有0和1，我们可以bfs，于是，做一遍复杂度与图的边数有关，即$O(8nm)$的。

750H. New Year and Snowy Grid

- 当然，继续观察可以发现，由于只要判断最短路与2的大小关系，所以问题就转化成：是否添加一个障碍物就可以使S到T的最短路为0，也就是，是否可以添加一个障碍物使得S和T联通。
- 关于联通性，很容易想到用并查集维护。
- 于是我们可以预处理出所有的障碍物形成的连通块，并预处理出所有相距为1的连通块对。
- 每次询问，把k个点周围的点用并查集连起来，最后看一下，所有与S并起来的连通块和所有与T并起来的连通块，是否存在连通块对相距为1，以及是否有新添的障碍点，满足这个点到S集的距离加上这个点到T集的距离不超过1。

772E. Verifying Kingdom

No.3

772E. Verifying Kingdom

- 【题目大意】交互题，交互库有一棵二叉树，这棵树的非叶节点的儿子树都为2。每次询问你可以给出(a,b,c)三个不同的点，交互库会回答你$\mathrm{lca}(a,b)$和$\mathrm{lca}(b,c)$和$\mathrm{lca}(a,c)$中深度最深的那个，若$\mathrm{lca}(a,b)$最深会返回字母X，若$\mathrm{lca}(b,c)$最深会返回字母Y，$\mathrm{lca}(a,c)$最深会返回字母Z。请你输出与交互库同构的任意一棵树。
- 【数据范围】$n \le 10^3$，询问不能超过$10n$

772E. Verifying Kingdom

- 考虑一个点一个点地加入树中，问题就转换成了如何快速判断一个点在当前树中的位置。
- 一个显而易见的想法是点分治。
- 每次找到当前树的重心，然后询问重心左子树的叶子、右子树的叶子和要加入的点。
- 如果返回的是左子树的叶子和加入点的lca最深，那么说明加入点在左子树，就点分左子树。右子树同理。
- 如果返回的是左右子树，那就说明要加入的点在外面，则点分外面的部分。
- 总的复杂度是$O(n^2 \log n)$。

627F. Island Puzzle

No.2

627F. Island Puzzle

- 【题目大意】有一棵$n+1$个点树，在上面玩n数码，问是否有解和最小步数，若无解，则问加一条边是否有解和最小步数。若不用添边则输出0和最小步数，若要添边则输出添的边和最小步数，若无解输出-1。
- 【数据范围】$n \le 2 \times 10^5$

627F. Island Puzzle

- 这道题其实不难，只是代码量比较大，特判比较多。
- 首先可以发现，如果空节点没有经过环的话（即不额外添边的情况），那么不管空节点怎么移动，只要空节点最后的位置确定，那么其它所有数字的位置都能确定。
- 而添一条边，就会得到环套树，那么空节点一定是先移动到环上，然后绕着环走若干圈，最后再走向目的地。
- 考虑如何判断-1的情况：先将空节点从起点移动到终点（设终点为end），看一下所有不在自己目标位置的数字是否可以添一条边组成一个环。如果不能组成，答案就是-1，如果所有数字都在目标位置上，那么，答案就是0，最小步数就是空节点从起点移到终点的距离。

627F. Island Puzzle

- 但是判断“所有不在自己目标位置的数字是否可以添一条边组成一个环”比较麻烦，需要具体分析一下：
- 如果只有一条链的话，假设数字组成的链的两端为x和y，且离end较近的那个点为x，设x'表示与x相邻的，且在路径(end, x)上的点，显然我们只能添(x', y)这条边；
- 如果有两条链的话（设其为l_1和l_2），必须存在一个点z，使得z与l_1的其中一个端点相邻，与l_2的其中一个端点相邻，并且z可以由空节点从end不经过这两条链到达，如果不满足的话，肯定无解。设离z距离不是1的l_1和l_2的端点分别为x和y，显然我们只能添(x, y)这条边。

627F. Island Puzzle

- 这样子，我们就知道了应该添的边是哪一条了，之后就是考虑最小步数的问题了。
- 最小步数一定是空节点走最短路到达环，再顺时针或逆时针绕环走，直到环上所有数字都到达了目标位置，最后走最短路到达终点。
- 整道题目的难度主要是在实现上，特判比较多。
- 总的时间复杂度，写的优美一点是$O(n)$的，写的暴力一点是$O(n\log n)$的，都可以通过本题。

725G. Messages on a Tree

No.1

725G. Messages on a Tree

- 【题目大意】一棵$n+1$个节点的有根树，编号从0到n，其中0号结点为根，父亲编号小于儿子编号，每条边进行信息传导都需要 1 秒。一共有m个操作，每个操作形如(x,t)，表示t时刻，节点x向它的父亲发了一条请求。对于一个收到请求的节点，如果是0号点或者等待节点，那么就直接发送回答给请求传来的节点，否则向父亲发送请求，并变成等待节点。对于一个收到回答的等待节点，会把回答发给请求传来节点，并解除等待。对于同一时刻收到多个请求的，先处理请求发起者编号小的请求，对于同时收到请求和回答的，先处理回答，再处理请求。你的任务是对于每个操作(x,t)，求出x收到自己发出请求的回答的时刻。
- 【数据范围】$n,m \le 2\times10^5$，$t \le 10^9$

725G. Messages on a Tree

- 显然应该把每个操作按时间与深度的和排序，这样子操作之间的优先级就确定了。
- 每个点维护一个等待结束的时刻，于是问题就变成了到根上路径求第一个大于x的点和连上覆盖等差数列。
- 就写写麻烦一点，思路还是挺清晰的，树剖+线段树，复杂度是$O(n\log^2 n)$。

THANKS

谢谢观看