Day 5 讲题

Special Judge

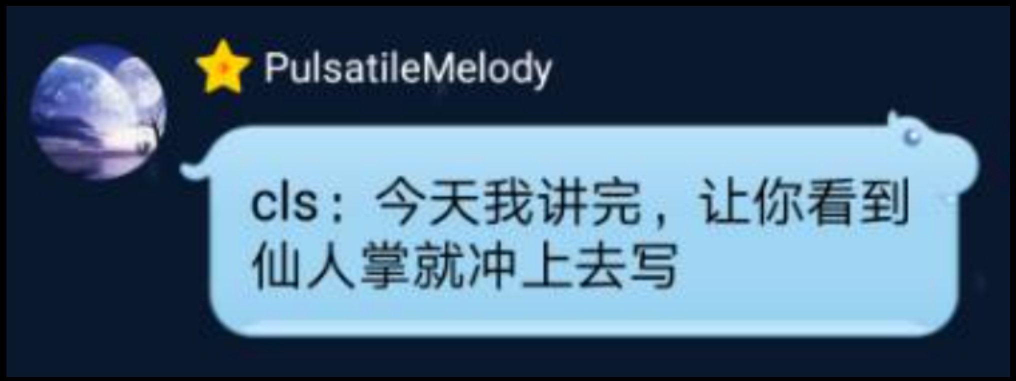
- 判有多少对线段相交。

Solution

- 如果不会做的同学可以打开Day5-div2-计算几何.pdf😍
- 没有公共端点就是不规范相交。
- 有就是判是不是同一个方向，用叉积+点积即可。

Cactus Draw

- 要求一个仙人掌的平面嵌入。
- Div 2：树



Solution

- 如果是树，那么按BFS顺序画一下就好了。

Solution

- DFS序，把每个环放在同一层。

Sorting

- 要求支持区间操作和区间求和。

Solution

- 容易发现<=x的数字和>x的数字的相对顺序不会变。
- 把<=x的数字当成0，>x的数字当成1。
- 操作就是区间求和和区间复制。
- 区间求和就相当于搞出这是第几个0到第几个0，第几个1到第几个1，然后前缀和即可。

Fast Kronecker Transform

- 求一个奇怪的卷积。

Solution

- 当数字出现次数少的时候，可以使用暴力。
- 出现次数多的时候可以使用FFT。
- 通过测试可以发现阈值设成T=10000的时候暴力和FFT跑的速度差不多。

Doppelblock

- 给一个矩阵，往里面填数字和X，满足一定的条件。
- 这个题Div 2过的比Div 1多。

Solution

- 搜索，加一点剪枝。
- 就过了。
- 标程先搜X的位置考虑a能用多少个数字拼出来剪枝。
- 然后再搜数字，用每个线索可行的数字集合剪枝。

Division

- 你有一个序列，对于一个区间，你可以做k次/2操作，问这些数字的和最小值。

- Div 2 : 一次询问。

Solution

- Div 2 : 每次减数字最大的，容易发现n * log W次之后所有数字都变成0了。

Solution

- 对于一个数字，求出每步操作的贡献。
- 答案就是区间最大的k个数字，可以用主席树解决。
- 但是直接用主席树做可能会有空间问题。
- 我们把所有的数字分到2^k到2^(k+1)分别做。
- 时间复杂度O(n log ^2 n + q log n)，空间O(n log n)。

Least Common Multiple

- 求[x+a1, x+a2, …, x+an]最多能除掉多大的因子。

Solution

- 首先当p整除于其中某两项的时候，才有可能会除掉。
- 所以p一定是某个ai-aj的素因子。
- 考虑素因子p，看能除掉多少个p。

Solution

- 假设lcm中p的指数是e次，并且在x+ai这个项取到。
- 记v(x)为x中p的幂次。
- 那么能除掉的个数是sum v(aj-ai)。
- 只需要枚举e，然后统计有多少个数字与ai模p^e同余即可。

Nested Tree

- 有一棵树，然后复制了若干份贴起来，求树上点两两之间的距离和。

- Div 2 : n, m <=1000

Solution

- Div 2 : 可以把整个树建出来，对于一条边，对答案的贡献是两端size乘积的和。

Solution

- 把所有树串起来之后，某些边的两端的大小会变。
- 我们可以用虚树/树链剖分解决。

Diameter

- 问n个点有标号的直径为0,1,…,n-1的树有多少棵。
- Div 2 : n<=11

Solution

- Div 2 : 可以通过Prufer序列枚举出所有子树。
- 大概30min就能算出答案。

Solution

- 首先考虑n个点的有标号生成树怎么计算。
- dp[n]表示n个点的有根树。
- 先不考虑根的标号，考虑除了根以外标号最小的节点所在的子树大小为k。
- dp[n]=n*sum dp[k]*dp[n-k]/(n-k)*C(n-2,n-k-1)

Solution

- 令dp[i][j][k]表示i个点的子树，最大深度为j，直径为k的方案数。

- dp[n][max(p1,p3+1)][max(p2,p4,p1+p3+1)]=n*sum dp[k][p1][p2]*dp[n-k][p3][p4]/(n-k)*C(n-2,n-k-1)

- 复杂度O(n^6)

Solution

- 每个树都有唯一的中心。
- 如果直径是偶数，那么中心是一个点，否则中心是一条边。
- 如果中心是一条边，那么要求两个子树的深度相同。
- 否则要求最大的深度出现了至少两次。
-

Solution

- 令f[i][j]表示i个点，深度至多为j的方案数。
- 令g[i][j]表示i个点，深度恰好为j的方案数。
- f[n][i]=n*sum f[n-k][i]/(n-k)*f[k][i-1]*C[n-2][k-1]
- g[n][i]=f[n][i]-f[n][i-1]
- 如果直径是2j+1，那么答案为sum g[k][j]*g[n-k][j]*C(n-1,k-1)
- 如果直径为2j，那么答案为g[n][j]-sum g[k][j-1]*f[n-k][j-1]*C(n,k)

Kropki

- 求1~n的排列个数，满足相邻两项是两倍或者不是两倍关系。
- Div 2 : n<=15

Solution

- Div 2 : 状压DP。

Solution

- 考虑容斥，将不能是两倍的情况变成一定要是两倍的情况。

- 那么这些数字形成层了若干条链。

- 考虑1-2-4-8 3-6-12-…也是若干条链的情况。

Solution

- dp的时候只记录链的长度，不需要记录顺序。
- 同理，dp出数字形成的链拆分的方案，还是只要记录顺序即可。
- 对于一堆链，通过简单组合计数能算出它们之间配对的方案数。
- 时间复杂度大概是O(P(n) * n^2)级别。