Count Angles

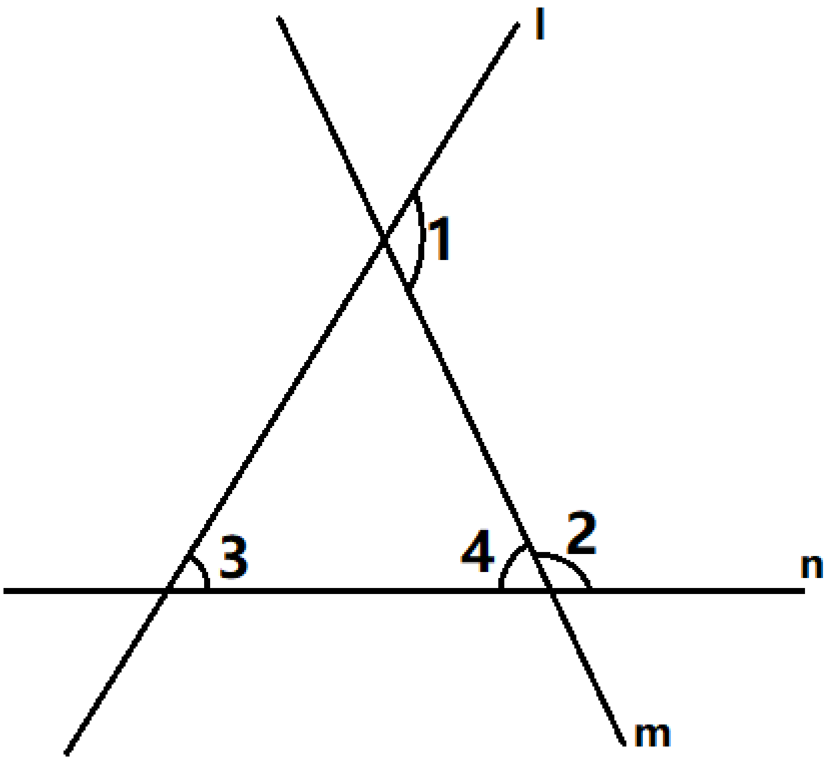
题目大意

T 组数据，每组数据给定 n。求平面上 n 条直线至多构成多少对同旁内角。答案对 10^9+7 取模。

数据规模：$1 \leqslant T \leqslant 10^5$，$1 \leqslant n \leqslant 10^9$。

解法

对于任意一对同旁内角，与其相关的直线有且仅有 3 条，而对于三条直线，他们至多形成 6 对同旁内角（如图，三条直线构成的三角形中内角三对、外角三对）。故当 n 条直线两两相交且不存在三线共点时，同旁内角对数最多，为 $6C_n^3$。



（如果通过规定三条直线的顺序来确定一对同旁内角的话，得到的答案是 A_n^3，如 lmn 确定 $\angle 1$ 和 $\angle 2$，lnm 确定 $\angle 3$ 和 $\angle 4$)

那么，“n 条直线两两相交且不存在三线共点”的情况一定存在吗？

一定存在。假设“$n-1$ 条直线两两相交且不存在三线共点”的情况存在，由于一定能找出一条直线与已知的 $n-1$ 条直线不平行，选取这条直线，则可以证明“n 条直线两两相交且不存在三线共点”的情况存在。

时间复杂度 $O(T)$。

另一种做法

本题也可以求出递推式，然后用矩阵快速幂去做。

所以在此讲讲递推的做法。

一种递推是：

设答案为 a_n。考虑在 n 条直线的基础上添加第 $n+1$ 条直线（设为 l），那么原来的 n 条直线构成 a_n 对与 l 无关的同旁内角，与 l 有关的同旁内角中，l 作截线的有 $2C_n^2$ 对，l 不作截线的有 $4C_n^2$ 对，所以得到递推式 $a_{n+1}=a_n+6C_n^2$。

考虑到 $C_{n+1}^2=C_n^1+C_n^2=n+C_n^2$，我们可以用矩阵来表示 a_n、C_n^2、n 的递推，进而用矩阵快速幂来快速计算答案。

（但是那位同学用的不是这个递推……）

时间复杂度为 $O(T\log n)$。

注意事项

- n 的值很大，所以要用 `long long`。
- n 的值很大，所以不要写成 $n*(n-1)*(n-2)\%1000000007$。

  真的有很多人这样 WA 了。。。

Grammy's Restaurant

被完全包含的区间价值恒 ≤ 包含他的区间的价值，因此去掉包含关系后，可以将所有区间排序，使得区间左右端点严格递增

有个性质是：对于左右端点同时递增的区间序列，他去覆盖一个长为n的数值序列，覆盖的数不同的区间只有O(n)种，每种对应区间序列上连续的一段。

对于覆盖的数相同的区间，价值最大的显然是最长的区间

于是对于每个询问就可以直接做了，具体实现有两种做法：

- 第一种是$O(nlogm)$的，对于数值序列的每个数，二分找到 第一个和最后一个覆盖了他的区间，然后可以在数值序列上维护一个双指针搞，询问一段区间最

长的可以用线段树或者 rmq

- 第二种是$O(nlogm(logn+logm))$的，由于每种对应连续的一段区间，先在区间序列上二分这一段的最右边那个区间，然后在数值序列上二分找到这个区间覆盖的数，来判断是不是同一种区间

数据没有卡 log^2 的做法，应该两种做法都能过

如果还有其他做法，欢迎分享

MUG

DP

If we end with the i-th point, we need to record the highest score of three condition.

$f(i,0)=max\{\ f(i-1,0),\ f(i-1,1),\ f(i-1,2)\}$

$f(i,1)=max\{\ f(i-1,0)+Si,\ f(i-1,1)+Si\times P3,\ f(i-1,2)+Si\}$ (when the i-th point is on the right)

or

$max\{\ f(i-1,0)+Si,\ f(i-1,1)+Si\times P3,\ f(i-1,2)+Si\times P2\}$ (when the i-th point is on the left)

$f(i,2)=max\{\ f(i-1,0)+Si,\ f(i-1,1)+Si\ \}$ (when the i-th point is on the left)

or

$max\{\ f(i-1,0)+Si,\ f(i-1,1)+Si\times P1\ \}$ (when the i-th point is on the right)

$f(0,0)=f(0,1)=f(0,2)=0$

The answer is $max\{\ f(n,0),\ f(n,1),\ f(n,2)\ \}$.

The time complexity is $O(N)$. The space complexity is $O(N)$ or $O(1)$.

Words

- 首先用容斥原理，可得由字符串集合A构成的Trie的节点数可以表示为$\sum_{S\in A}(-1)^{|S|+1}\mathrm{LCP}(S)$，因为字符串是均匀随机的，所以$\mathrm{LCP}$只和字符串数量有关，所以我们直接枚举数量$\sum_{i=1}^{n}C_n^i(-1)^{i+1}f(i)$，其中$f(i)$代表$i$个满足上述条件的串的$\mathrm{LCP}$的期望，这个式子是可以直接$\mathrm{NTT}$的，$A(x)=\sum_{i=1}^{n}\frac{(-1)^i f(i)}{i!}x^i, B(x)=\sum_{i=0}^{n}\frac{1}{i!}x^i, C(x)=A(x)B(x)$，接下来考虑怎么求$f(i)$，我们枚举第$j$个字符出现在在$\mathrm{LCP}$中的期望，那么固定一个字符串后，剩下的$i-1$个字符串只要满足前$j$位都和第一个相同即可，所以期望是$(\frac{1}{c})^{(i-1)j}$，那么$f(i)=\sum_{j=1}^{L}(\frac{1}{c})^{(i-1)j}$，$i\neq 1$时，这个是等比数列求和，那么可以O$(\log L+\log c)$求，所以复杂度就是O$(n(\log n+\log L+\log c))$
- 把问题看成一棵L层的完全c叉树，从根走到叶子，一共出发n次，求经过的节点个数期望，那么我们按层枚举所有节点，计算至少到达一次该节点的概率，也就是完全不到达该节点的概率的补集，那么就是$\sum_{i=1}^{L}c^i(1-(1-\frac{1}{c^i})^n)$，考虑求$\sum_{i=1}^{L}c^i(1-\frac{1}{c^i})^n$，二项式展开，$\sum_{i=1}^{L}c^i\sum_{j=0}^{n}(-1)^j(\frac{1}{c^i})^j$，交换一下求和顺序就可以得到和另一种做法一样的式子

气球塔防

- 给n个点，求是否存在一条直线上点数大于总点数的p%
- 保证p不太小

- 做法:
- 重复多次
- 每次随机找两个点，判断两点连线上点数是否大于总点数的p%
- 如果答案是possible，那么每次有一个不太小的概率使得随机的两个点都在这条直线上，因此重复多次有很大概率正确
- 时间复杂度 $O(Tn)$

几乎相邻

- 空图，多次操作，每次加边或询问两点距离是否恰好为2

- 暴力做法：
- 哈希表维护邻接表，每次for一for

- 正解做法：
- 以下称度数小于$\sqrt{m}$的点为小点，其余点为大点
- 询问时有一个小点则暴力for小点的邻接表查大点是否与对应的点相邻
- 在维护一个$\sqrt{m} \times \sqrt{m}$的表格表示所有两个大点形成的点对之间的答案，一边加边一边更新
- 小点变成大点时for一遍这个点的邻接表，更新它和所有大点之间的答案（昨天晚上刚发现标程没写这个）

- 时间复杂度$O(m\sqrt{m})$，空间复杂度$O(n+m)$