

T1

对于每一个点计算一下距离+的距离，可以用bfs实现

计算两点之间经过点最大值最小的可以用二分答案+bfs或者用dijkstra来实现

$O(nm\log)$

T2

容易发现这个操作其实就是手里初始有一个数，每次能和其中一个位置交换，最后要 A, B 相同

那么我们发现对于每一个联通块需要多操作一次，维护一下构成了几个联通块即可

由于数值范围比较大，需要先离散化一下

$O(n \log n)$

T3

$O(N \log_2 N)$

首先考虑以 i 为根的答案，可以发现，如果 (j, k) 没有祖先关系，贡献是 $1/2$ ，假如 (j, k) 有祖先关系， j 是 k 的祖先，我们可以通过dfs统计 $j > k$ 的对数，贡献是1，即dfs的时候维护树状数组，设 $f[i]$ 为 i 为根的有根树的子树逆序对数

然后考虑 i 转移到 $son[i]$ (i 的某个孩子)，有 $f[son[i]] = f[i] - i$ 对 $son[i]$ 这棵子树贡献的逆序对数 + $son[i]$ 对除 i 这棵子树贡献的逆序对数

可以发现，只有这两个节点的 f 值会改变，其他点都不会改变，因为只有他们变换了父子关系，其他都是兄弟关系

然后我们发现可以维护 i 对 $son[i]$ 这棵子树的逆序对(dfs序+主席树)/线段树合并

因此我们可以 $\log n$ 的从 i 的答案转移得到 $son[i]$ 的 f 值

$O(N^2)$

考虑一个数对 (j, k) 的贡献，假设 $j < k$ ，可以发现，

如果从 k 的子树开始遍历(必定先经过 k)，那么他们吃不到这个逆序对

假设从 j, k 中的某个点开始遍历(可以先 j 也可以先 k)，逆序对的期望数是 $1/2$

假设从 j 以外的节点开始遍历(必定先经过 j 的点)，逆序对数的期望是1

于是我们从 j 开始搜索，如果搜索到 $k > j$ 就给 j 打一个 $+1/2$ 的标记，再给 j 本身 $-1/2$ (因为 j 本身吃不到)，再给 k 打一个 $+1/2$ 的标记，最后下放节点即可，更新节点值即可

u相等怎么使用

先考虑这个题：1... n 的排列的期望逆序对是多少

答案是 $\frac{C_n^2}{2}$ ，因为一组数 (a, b) ，要么 a 在 b 前要么 b 在 a 前，他们的贡献是 $\frac{1}{2}$

可以发现只有 u 的位置是固定的(首位)，其他的数可以在dfs序中2.. n 的任何一个位置

也就是说 u 这个点的答案是 $\frac{C_{n-1}^2}{2}$

然后考虑其他的点

我们可以发现点 x 为根的时候他肯定在 u 前面，然后其他的数肯定是在dfs序中3.. n 中的任何一个位置

也就是说其他点的答案是 $\frac{C_{n-2}^2}{2} + [x > u]$

$O(N)$

一条链怎么使用

当 x 为根的时候我们可以发现 x 肯定在最前面，但是除此之外，只有 x 左边的链和 x 右边的链产生的贡献，

也就是说我们可以有 x 左边的数* x 右边的数这么多对贡献为 $1/2$ 的(跟菊花图一个道理)

可以树状数组 $O(N \log_2 N)$ 正着扫一遍反着扫一遍来实现

$O(N \log_2 N)$

T4

10pts

$O(2^{16} * 16)$ 暴力枚举

30pts

注意到原图其实被分为了两个图(根据 $i + j$ 的奇偶性)

根据观察能发现有一些点不可能被选取

于是只需要暴力枚举可能选择的点是否选到

$O(2^{15} * 15)$

50pts

进一步观察原图性质

我们会发现将图进行旋转之后题目限制可以变成要求 $(i - 1, j)$ 和 $(i - 1, j + 1)$ 被选

进一步转化可以发现问题变成每一列上要选取一个前缀, 并且当前列前缀高度小于等于右边这一列高度+1

这个显然可以利用 dp 完成

时间复杂度 $O(n^5)$, 可以优化到 $O(n^4)$

80pts

有一个点可以不符合题目限制将问题复杂化了一些

可以分成三种情况

1. 选一个和当前列已选前缀无关的点 (就是在下面)
2. 使得当前列高度可以等于右边一列高度+2
3. 这种情况比较容易遗漏, 当前列可以在前缀中挖空一段(满足一定条件)

前两种情况相较于原本的 dp 没有什么太大变化

第三种情况需要增加枚举挖空点的上下边界

注意做完 dp 后还要考虑上对角线的特殊情况

时间复杂度 $O(n^6)$

100pts

注意到第三种情况其实不用枚举上下边界

只需要枚举下边界后用前缀和优化

时间复杂度优化到 $O(n^5)$

但这样可能仍无法通过

注意到原图的一些位置是无法选取的, 所以在 dp 的时候把这些状态去掉, 可以除去很大的常数