solution

小β送快递（delivery）

签到题直接上正解。

考虑树形DP，dp_i表示以i为根的子树的所有除了i以外所有节点的答案，与从根节点直接到i后i的答案取最大值。

考虑对于节点i的所有子节点v，由于需要一个子树一个子树访问，考虑怎么样才能使最大的满意时间最小。

容易发现，按照$2(size_v-1)+dp_v$为关键值对子树排序，从小到大访问即可。

答案即为$max(dp_1,val_1+2*(n-1))$

时间复杂度$O(nlogn)$

小 o 数排列（bright）

阳光开朗的小 o 和大家问好。

首先观察到 4 只有除 1 外所有都逆序的情况，可以输出 1 获得 5 分的好成绩。

观察到 $5,6,7,8$ 四个点固定的数并不会对构成递增三元组造成什么影响。先考虑没有固定数的做法。

第 4 个点虽然是用来送分的，但也给了我们一些启发，发现拥有三个递增必须先有两个递增。

考虑从小到大加数。加数时记录位置最前的两个递增的位置 p （也就是第一次出现非完全倒序的地方），当加入当前数 i 时，若 i 插在 p 的后面，则会形成满足条件的三元组，所以 i 只能插在 $1-p$ 的任意一个位置中，若插在位置 1 中，则 $dp[p]$ 转移到 $dp[p+1]$ ，若插在 $2-p$ 中的其中一个位置 $q\le p$ 则由 $dp[p]$ 转移到 $dp[q]$ 。

然后对于没有限制数的即可得到一个 $O(n^3)$ 的dp，发现转移是一个后缀区间，即可使用后缀和优化变为 $O(n^2)$。

现在考虑有限制数的情况，当加入 x 时，在原来的 dp 上加上一维 $dp[posx][p]$ ，插入时讨论一下 $posx$ 是否需要后移一位，按照原方法做 dp 即可，时间复杂度为 $O(n^3)$。

小ρ修城市（city）

签到题直接上正解。

容易发现这是一个矩阵树定理的题。

观察其删去第一行第一列的代数余子式可得，行列式只有主对角线是某些值，以及主对角线两侧的对角线为1，其余都是0。

不难想到行列式按行展开，用ans_i表示该代数余子式的前i行i列的值，

则$ans_i = ans_{i-1}val_{i,i} - a_{i-2}$

其中$val_{i,i}$表示主对角线从左上到右下第i个数的值。

输出$n-1$即可。

时间复杂度为$O(Tn)$。

小 χ 的星系（galaxy）

出题人由于能力不够脑子有些混所以在不清醒的情况下出了一道自己也不会的题，有些影响欢乐赛的体验，万分抱歉，感谢 mrsrz 提供的 std 和 sol。

首先考虑对于第 0 天如何计算。

考虑其中一个点对 $(i,j)(i<j)$ 贡献。

会出现两种情况：

- i,j 同时在随机交换区间内。重排后会随机打乱，那么 (i,j) 形成逆序对的概率即为 0.5
- i,j 不同时在随机交换区间内。那么 (i,j) 的相对位置不会改变，只要关注原来的顺序即可。

从前往后处理，在第 i 个位置时：

- 第一种情况的贡献为：$\frac{1}{2}\sum_{j=1}^{i-1}(n-i+1)\times j=\frac{1}{4}(n-i+1)\times i\times(i-1)$
- 第二种情况的贡献为：$\sum_{j=1}^{i-1}[p[i]>p[j]](tot-j\times(n-i+1))$，即用总区间个数减去两者在同一个区间内的情况数。

其中 $tot=\frac{1}{2}n\times(n+1)$ 代表总区间数。

对于第二种情况，只需用两个树状数组维护 $p[i]>p[j]$ 的个数即下标和即可。

对于每次询问用此方法计算就能获得很高的分数。

下面是对于修改部分的处理。

大家好这里是负责背 std 锅的工具人。

我们容易发现，现在只需要维护这样两个东西：

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}[p[i]>p[j]]$$

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}[p[i]>p[j]]\cdot i\cdot(n-j+1)$$

需要支持单点修改。

第一个东西就是在维护逆序对，容易使用树套树/CDQ 分治来解决。

第二个东西也可以用类似的方式来维护。

我们考虑位置 x 被换到 y（设 $x<y$）。那么：

- 对于 $1\le i<x$ 的所有满足 $p_i>p_x$ 的 i，都要减去 $i\times(n-x+1)$，再加上 $i\times(n-y+1)$。
- 对于 $y<i\le n$ 的所有满足 $p_x>p_i$ 的 i，都要减去 $x\times(n-i+1)$，再加上 $y\times(n-i+1)$。
- 对于 $x<i<y$ 的所有满足 $p_x>p_i$ 的 i，都要减去 $x\times(n-i+1)$。
- 对于 $x<i<y$ 的所有满足 $p_x<p_i$ 的 i，都要加上 $y\times(n-i+1)$。

这些东西似乎较难用 CDQ 分治求（也许可以），但是用树套树是可以维护的。只需要维护两棵树套树，分别维护点 i 贡献为 i 和贡献为 $(n-i+1)$ 时的两棵树套树，查询时还是一个二维数点问题。时间复杂度 $O(q\log^2 n)$，空间复杂度根据套法不同为 $O(n\log n)$ 或 $O(n\log^2 n)$。估计 $O(n\log n)$ 空间的常数非常好的树套树可以通过本题。

~~出题人的 std 因为使用太多的空间所以死了所以我不是出题人~~

由于树套树常数太大，所以考虑选择树套树以外的方式维护这些东西。

对于本题，常数优秀的根号算法是可以接受的。

介绍两种做法：

1. 对于 $x < i < y$ 的情况可以和前面的情况合并，这样我们的查询就只有前、后缀查询了。

   仅对前缀查询进行考虑，即单点修改、前缀查询。这是一个二维的问题，可以使用莫队套值域分块的做法进行求解。

   时间复杂度 $O(q\sqrt{n} + n\sqrt{q})$。空间复杂度 $O(n + q)$，需要离线处理询问。

2. 序列分块+值域分块（std 采用的算法）。

   考虑对序列进行分块，每个块内再对值域分块。同样对于 i 和 $(n - i + 1)$ 的贡献分开记录。

   我们用 $b_{i,j}$ 记录第 i 个序列块中，前 j 个值域块记录的贡献和，即块前缀和数组。

   对每次交换 x, y，我们枚举所有不包含 x, y 的块。然后再用前缀和信息，就可以 $O(1)$ 计算这个序列块中的数与 x, y 的贡献。算上枚举块的复杂度，时间复杂度为 $O(\sqrt{n})$。

   考虑这样处理完后，还有哪些东西的贡献没有计算：

   x 所在序列块中的其他数对 x, y 的贡献，y 所在序列块中的其他数对 x, y 的贡献；p_x 所属值域块对 x 的贡献，p_y 所属值域块对 y 的贡献。这里最多要新算 $O(\sqrt{n})$ 个数的贡献。

   于是单次更新的复杂度为 $O(\sqrt{n})$。

   交换两个数后，还要更新分块记录的信息。我们只需要更新 x, y 所在块的前缀和信息即可，所以单次更新也是 $O(\sqrt{n})$ 的。

   为了方便，逆序对个数也可以同时分块维护。

   总时间复杂度为 $O(q\sqrt{n})$，空间复杂度 $O(n)$，可以在线处理询问。

时空限制还是比较松的所以 $O(n\sqrt{n})$ 空间的做法和 $O(q\sqrt{n \log n})$ 时间的做法大概也能过。