seq

对于 $n \le 15$，枚举每个子序列然后排序计算即可。时间复杂度 $\mathcal{O}(2^n n\log n)$。

对于 A_i 互不相同，可以枚举每个元素的贡献。即若 A_i 满足在某一子序列中排名第 A_i，则有 1 的贡献。也就是当 $1 \sim A_i$ 都被选择时才能有贡献。而大于 A_i 的数选不选都可以，所以贡献为 2^{n-A_i}。

发现可以将原序列直接排序不影响答案结果，那么还是考虑每个元素的贡献求和。对于 A_i，需要在前面 $i-1$ 个数中选取恰好 A_i-1 个，后面 $n-i$ 个数选不选都可以，预处理阶乘快速算组合数即可。时间复杂度 $\mathcal{O}(n)$。

destroy

摧毁最多的道路就对应着保留最少的路径。当 $s_1=s_2,t_1=t_2$ 时，直接保留最短路即可。

所以答案上界是 $dis(s_1,t_1)+dis(s_2,t_2)$，其中 $dis(x,y)$ 表示 x 到 y 的最短路长度。

但是存在两条路径会重叠的情况，重叠情况只需要被计算一次。

由于 n,m 只有 3000，所以可以通过枚举两个最短路重叠部分。设枚举的重叠部分端点为 a,b，那么答案为 $\min_{(a,b)}\{dis(a,b)+\min(dis(s_1,a)+dis(b,t_1),dis(t_1,a)+dis(b,s_1))+\min(\text{这里和前一个式子类似})\}$.

所以需要处理任意两点最短路，使用 BFS 即可。时间复杂度 $\mathcal{O}(n(n+m))$。

tree

模拟题意暴力比较两两字典序，复杂度 $\mathcal{O}(qn^2)$。

首先 L 至少是 P_1 的一个前缀。否则会有 $L>P_1$，不满足 L 是字典序最小的条件。在 L 是 P_1 前缀的前提下，$|L|$ 越小，则 L 字典序也越小。

当 a_i 是排列的情况下，若 $sz_x=k$，则要求 $1\sim k$ 全部在 sz_x 的子树内。这里给出一种做法：对于 $i\in[1,n-1]$，设 $a_x=i,a_y=i+1$，对 (x,y) 路径上除了 y 的点执行 $+1$ 操作。那么上述要求等价于 x 的权值等于一。树剖线段树维护子树最小值（第一关键字按权值排序，第二关键字按 sz 排序）即可。时间复杂度 $\mathcal{O}(n\log^2 n)$。

当 a_i 不是排列时，就无法用上述方法了。我们需要找到更简便的，描述前缀这个限制：定义 d_u 表示 u 当前子树和与满足条件的子树和之差。也就是说 $d_u=\sum_{v\in sub(u)}a_v-\sum_{i=1}^{sz_u}P_{1_i}$。

于是问题转化为需要找到 sz 最小的，且 $d=0$ 的点。同样用树剖线段树来实现，时间复杂度 $\mathcal{O}(n\log^2 n)$

divide

考虑给定一个 a 如何计算 $F(a,d)$。

首先将 a 排序，可知将值域连续的分成一段不会更劣。所以最终分成 L 个子集可以看作分成 L 个子段。

考虑 $L=2$ 的情况，本来是 a_n-a_1+d ，假设将前 i 个分为一段，那么代价变成了 $a_i-a_1+a_n-a_{i+1}+2d$ 。容易发现就是去掉了 $a_{i+1}-a_i$ 在加上了一个 d 。这启发我们考虑差分数组。

定义 $a_0=0,a_{n+1}=m$，设 $b_i=a_i-a_{i-1}$ ，则集合 a 可以序列 b 唯一确定。此时需要对 b 有些约束：$b_i\ge 0$ 且 $b_1\ge 1$；$\sum_{i=1}^{n+1} b_i=m$。（如果不加 b_{n+1} 需要花费 $\mathcal{O}(m)$ 的代价枚举总和。）

用 $\sum b_i=m-1$ 和 $b_i\ge 0$ 来替换以上约束。

对于一个由序列 b 构成的 a ，$F(a,d)=d+\sum_{i=2}^{n}\min(d,b_i)$。

不妨先让 $k=k-d$。

此时暴力 dp 有 $f(i,j,c,d)$ 表示当前处理 $b=i$ 情况，已经选了 j 个，代价和为 c ，数字和为 d ，转移枚举选了几个。根据不同优化拿到不同分数。

对于 $b_i<d$ 和 $b_i\ge d$ 的元素我们需要分别处理。假设有 x 个 $b_i\ge d$ ，首先有 $\binom{n-1}{x}$ 种方案固定位置。这些位置可以看作 $b_i=d+c_i$ ，其中 $c_i\ge 0$。

$$
\begin{gathered}
b_1+b_2+\ldots+b_{n+1}=m-1\\
b_1+b_{n+1}+(b_2+b_3+\ldots+b_{x+1})+(b_{x+2}+\ldots+b_n)=m-1\\
b_1+b_{n+1}+x\cdot d+(c_2+c_3+\ldots+c_{x+1})+(k-x\cdot d)=m-1\\
b_1+b_{n+1}+c_2+c_3+\ldots+c_{x+1}=m-1-k
\end{gathered}
$$

所以用插板法计算下面的式子即可。而另一部分要求：

$$
\begin{gathered}
b_{x+2}+\ldots+b_n=k-xd\\
0\le b<d
\end{gathered}
$$

使用容斥即可 $\mathcal{O}(n)$ 解决。具体地，钦定有 i 个 $b\ge d$，用插板法计算答案，乘上$(-1)^i$ 相加。

这部分如果用背包预处理，即 $f_{i,j}$ 表示 i 个不超过 d 的值相加等于 j 的方案数，可以在 $\mathcal{O}(nm)$ 的复杂度内预处理得到，然后 $\mathcal{O}(1)$ 查询。

最后总时间复杂度 $\mathcal{O}(n^2)$。