冒泡排序（inverse）

【题目背景】

最近，小 S 对冒泡排序产生了浓厚的兴趣。为了问题简单，小 S 只研究对 1 **到 n 的排列**的冒泡排序。

下面是对冒泡排序的算法描述。

```
输入：一个长度为 n 的排列 p[1...n]
输出：p 排序后的结果。
for i = 1 to n do
    for j = 1 to n - 1 do
        if(p[i] > p[i + 1])
            交换 p[i] 与 p[i + 1] 的值
```

冒泡排序的交换次数被定义为交换过程的执行次数。可以证明交换次数的一个下界是 $\frac{1}{2}\sum_{i=1}^{n}|i-p_i|$，其中 p_i 是排列 p 中第 i 个位置的数字。如果你对证明感兴趣，可以看提示。

【题目描述】

小 S 开始专注于研究长度为 n 的排列中，满足交换次数 $=\frac{1}{2}\sum_{i=1}^{n}|i-p_i|$ 的排列（在后文中，为了方便，我们把所有这样的排列叫“好”的排列）。他进一步想，这样的排列到底多不多？它们分布的密不密集？

小 S 想要对于一个给定的长度为 n 的排列 q，计算字典序严格大于 q 的“好”的排列个数。但是他不会做，于是求助于你，希望你帮他解决这个问题，考虑到答案可能会很大，因此只需输出答案对 998244353 取模的结果。

【输入格式】

从文件 ***inverse.in*** 中读入数据。

输入第一行包含一个正整数 T，表示数据组数。

对于每组数据，第一行有一个正整数 n, 保证 $n \le 6 \times 10^5$。

接下来一行会输入 n 个正整数，对应于题目描述中的 q_i，保证输入的是一个 1 到 n 的排列。

【输出格式】

输出到文件 ***inverse.out*** 中。

输出共 T 行，每行一个整数。

对于每组数据，输出一个整数，表示字典序严格大于 q 的“好”的排列个数对 998244353 取模的结果。

【样例 1 输入】

```
1
3
1 3 2
```

【样例 1 输出】

```
3
```

【样例 1 解释】

字典序比 1 3 2 大的排列中，除了 3 2 1 以外都是“好”的排列，故答案为 3。

【样例 2 输入】

```
1
4
1 4 2 3
```

【样例 2 输出】

```
9
```

【样例 3】

见选手目录下的 *inverse/inverse3.in* 与 *inverse/inverse3.ans*。

【子任务】

下面是对本题每个测试点的输入规模的说明。

对于所有数据，均满足 $T = 5$ (样例可能不满足).

记 n_{max} 表示每组数据中 n 的最大值，$\sum n$ 表示所有数据的 n 的和。

<table>
<tr><th>测试点</th><th>$n_{max} =$</th><th>$\sum n \le$</th><th>特殊性质</th></tr>
<tr><td>1</td><td>8</td><td rowspan="11">$5n_{max}$</td><td rowspan="11">无</td></tr>
<tr><td>2</td><td>9</td></tr>
<tr><td>3</td><td>10</td></tr>
<tr><td>4</td><td>12</td></tr>
<tr><td>5</td><td>13</td></tr>
<tr><td>6</td><td>14</td></tr>
<tr><td>7</td><td rowspan="2">16</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td><td>17</td></tr>
<tr><td>10</td><td rowspan="2">18</td></tr>
<tr><td>11</td></tr>
<tr><td>12</td><td>122</td><td rowspan="5">700</td><td>$\forall i \quad p_i = i$</td></tr>
<tr><td>13</td><td>144</td><td rowspan="4">无</td></tr>
<tr><td>14</td><td>166</td></tr>
<tr><td>15</td><td>200</td></tr>
<tr><td>16</td><td>233</td></tr>
<tr><td>17</td><td>777</td><td rowspan="4">4000</td><td>$\forall i \quad p_i = i$</td></tr>
<tr><td>18</td><td>888</td><td rowspan="3">无</td></tr>
<tr><td>19</td><td>933</td></tr>
<tr><td>20</td><td>1000</td></tr>
<tr><td>21</td><td>266666</td><td rowspan="5">2000000</td><td>$\forall i \quad p_i = i$</td></tr>
<tr><td>22</td><td>333333</td><td rowspan="4">无</td></tr>
<tr><td>23</td><td>444444</td></tr>
<tr><td>24</td><td>555555</td></tr>
<tr><td>25</td><td>600000</td></tr>
</table>

【提示】

下面是对交换次数下界是 $\frac{1}{2}\sum_{i=1}^{n}|i-p_i|$ 的证明。

排序本质上就是数字的移动，因此排序的交换次数应当可以用数字移动的总距离来描述。对于第 i 个位置，假设在初始排列中，这个位置上的数字是 p_i，那么我们需要将这个数字移动到第 p_i 个位置上，移动的距离是 $|i-p_i|$。从而移动的总距离就是 $\sum_{i=1}^{n}|i-p_i|$，而冒泡排序每次会交换两个相邻的数字，每次交换可以使移动的总距离至多减少 2。因此 $\frac{1}{2}\sum_{i=1}^{n}|i-p_i|$ 是冒泡排序的交换次数的下界。

并不是所有的排列都达到了下界，比如在 $n=3$ 的时候，考虑排列 3 2 1, 这个排列进行冒泡排序以后的交换次数是 3，但是 $\frac{1}{2}\sum_{i=1}^{n}|i-p_i|$ 只有 2。