Dispatch Money

We need to divide the permutation into subsegments, where the cost of the subsegment is equal to $x +$ the number of inversions on this subsegment.

Let's denote $f_{l,r}$ as the number of inversions on the segment.

Note that $f_{l,r} = f_{l+1,r} + f_{l,r-1} - f_{l+1,r-1} +$ (1, if $p_l > p_r$).

So $f_{l,r} + f_{l+1,r-1} \geq f_{l+1,r} + f_{l,r-1}$, that means that we can use some DP optimizations to solve the problem.

We need to optimize $dp_i = \min(dp_j + f_{j+1,i})$.

We will calculate this DP using the Divide and Conquer method.

At first, let's calculate DP values for $1, \ldots, \frac{n}{2}$ (recursively).

Then, we should proceed with DP transitions from $i \leq \frac{n}{2}$ to $j > \frac{n}{2}$, and then, we should calculate DP values for $\frac{n}{2} + 1, \ldots, n$.

Note that because of the property of our function, the optimal $i \leq \frac{n}{2}$ for j's are monotone.

So we can use the Divide and Conquer for monotone functions, we can find the optimal point for $i = \frac{(l+r)}{2}$ and then proceed with the smaller segments of candidates to the optimal points to the left and the right.

But how to calculate $f_{l,r}$?

During Divide and Conquer, we need to move l and r only $\mathcal{O}(n \log n)$ total times, so we can maintain $f_{l,r}$ and change it when we need to decrease/increase l/r. We can do it using BIT, to get **very** fast $\mathcal{O}(n \log^3 n)$ solution.

Also, it is possible to precalculate some information using SQRT decomposition, because we can note that our query is decomposed into some number of queries of the form "`find the number of integers` $\leq x$ `on the segment` $l \ldots r$", using sqrt decomposition we can calculate these values in $\mathcal{O}(1)$ with $\mathcal{O}(n\sqrt{n})$ precalc, to get $\mathcal{O}(n \log^2 n + n\sqrt{n})$ solution.