Permutations and Queries

Problem author: Alexey Mikhnenko

Problem developer: Alexey Vasiliev

Let us represent a permutation as a set of points (i, p_i). Then the first type of operation transforms all points (x, y) into $(n - x + 1, y)$. The second type of operation transforms (x, y) into $(x, n - y + 1)$. The third type of operation transforms (x, y) into (y, x).

Notice that using any number of these operations, we can obtain no more than eight different permutations, which we describe as follows. Let f_1, f_2, f_3 be three variables that take values 0 or 1.

Let p be our initial permutation, and then we apply the following modifications to it in order:

1. If $f_1 = 1$, then we perform the third type of operation (that is, we transform (x, y) into (y, x)).
2. If $f_2 = 1$, then we perform the first type of operation (that is, we transform (x, y) into $(n-x+1, y)$).
3. If $f_3 = 1$, then we perform the second type of operation (that is, we transform (x, y) into $(x, n-y+1)$).

Initially, we precompute the cost of the eight permutations corresponding to every possible set of variables f_1, f_2, f_3.

Now we maintain the current variables f_1, f_2, f_3, which are initially equal to zero.

- For a query of the first type, we replace f_2 with $1 - f_2$.
- For a query of the second type, we replace f_3 with $1 - f_3$.
- For a query of the third type, we replace f_1 with $1 - f_1$ and swap f_2 and f_3.

Thus, after each query, we maintain the current variables f_1, f_2, f_3 and output the precomputed cost for such flags. The running time is $O(n \log n + q)$, where the log comes from binary exponentiation used to compute the cost.