Review of Bubble Sort 2

Adding (small value)$\times i$ to the i-term of the sequence, we may assume all the values of the sequence are different from each other.

For each element in the sequence, consider the set $\{$ elements which are prior to it and larger than it $\}$. If this set is non-empty, after a pass, the element will move one step forward. Therefore, the maximum value of $\{$ elements which are prior to it and larger than it $\}$ will be the number of passes by bubble sort. They can take the maximum value only at vertices in the right lower convex hull of the sequence. For these vertices, the equality

$$\begin{aligned}&(\text{the number of elements which are prior to it and larger than it})\\&\quad = i - (\text{the number of elements smaller than } A_i)\end{aligned}$$

holds. For other vertices, the inequality

$$\begin{aligned}&(\text{the number of elements which are prior to it and larger than it})\\&\quad > i - (\text{the number of elements smaller than } A_i)\end{aligned}$$

holds. Therefore, the number of passes by bubble sort for the sequence A is calculated as

$$\max\left\{ i - (\text{the number of elements smaller than } A_i) \;\middle|\; i = 0, 1, \cdots, N-1 \right\}.$$

It is enough to keep the value of

$$i - (\text{the number of elements smaller than } A_i)$$

for each i, and to update and calculate the maximum value efficiently. Since the term i does not change, we may calculate it simply for each query. In order to update the term

$$-(\text{the number of elements smaller than } A_i)$$

for each query, we may add $+1$ or -1 to every element in the segment containing A_i.

Using Segment Tree, the above operations can be done $O(\log(N+Q))$ time per query. It takes $O((N+Q)\log(N+Q))$ time to compress the coordinates which appear in the sequence. Therefore, in total, this problem can be solves in $O((N+Q)\log(N+Q))$ time.