CodeForces Educational Round 178 E

omeganot

August 3, 2025

For an arbitrary k, when we move an element to the end there are two cases.

The first is when the element we move to the end is already one of the last k elements. In this case, the sum doesn't change.

On the other hand, if we move a different element to the end, then if previously, the last k elements were $a_{n-k+1}, a_{n-k+2}, \ldots a_n$, they are now $a_{n-k+2}, a_{n-k+3}, \ldots x$, where x is the value of the element we moved to the end. As such, the sum increases by $x - a_{n-k+1}$. If $x > a_{n-k+1}$, our sum increases, so we always want to do this operation if we have a big enough element to move to the end.

This motivates us to compute p_i, the maximum element of the prefix ending at i. In other words, we precompute the $\max(a_1, a_2, \ldots, a_i)$ for all i. We can use the fact that $p_i = \max(p_{i-1}, a_i)$ to compute this in $O(n)$. Then, for each k from 1 to n, we maintain the sum of the last k elements.

All we have to do is check if $p_{n-k} > a_{n-k+1}$ (if $k = n$, the answer is just our sum). If $p_{n-k} > a_{n-k+1}$, then we should move the element with value p_{n-k} to the end and our answer is $s + p_{n-k} - a_{n-k+1}$. Otherwise, our answer is just s. Here, we use s as the sum of the last k elements. In my code, I iterate from $i = n - 1$ to $i = 0$, where $i = n - k$. We solve the problem in $O(n)$.