A non-empty zero-indexed array A consisting of N integers is given.

A *monotonic_pair* is a pair of integers (P, Q), such that 0 ≤ P ≤ Q < N and A[P] ≤ A[Q].

The goal is to find the monotonic_pair whose indices are the furthest apart. More precisely, we should maximize the value Q − P. It is sufficient to find only the distance.

For example, consider array A such that:

```
    A[0] = 5
    A[1] = 3
    A[2] = 6
    A[3] = 3
    A[4] = 4
    A[5] = 2
```

There are eleven monotonic_pairs: (0,0), (0, 2), (1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (3, 3), (3, 4), (4, 4), (5, 5). The biggest distance is 3, in the pair (1, 4).

Write a function:

```
int solution(int A[], int N);
```

that, given a non-empty zero-indexed array A of N integers, returns the biggest distance within any of the monotonic_pairs.

For example, given:

```
    A[0] = 5
    A[1] = 3
    A[2] = 6
    A[3] = 3
    A[4] = 4
    A[5] = 2
```

the function should return 3, as explained above.

Assume that:

- N is an integer within the range [1..300,000];
- each element of array A is an integer within the range [−1,000,000,000..1,000,000,000].

Complexity:

- expected worst-case time complexity is O(N);
- expected worst-case space complexity is O(N), beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.