

## 962. Maximum Width Ramp

A ramp in an integer array `nums` is a pair  $(i, j)$  for which  $i < j$  and `nums[i] ≤ nums[j]`. The width of such a ramp is  $j - i$ .

Given an integer array `nums`, return *the maximum width of a ramp* in `nums`. If there is no ramp in `nums`, return 0.

### Example 1:

**Input:** `nums = [6,0,8,2,1,5]`

**Output:** 4

**Explanation:** The maximum width ramp is achieved at  $(i, j) = (1, 5)$ : `nums[1] = 0` and `nums[5] = 5`.

### Example 2:

**Input:** `nums = [9,8,1,0,1,9,4,0,4,1]`

**Output:** 7

**Explanation:** The maximum width ramp is achieved at  $(i, j) = (2, 9)$ : `nums[2] = 1` and `nums[9] = 1`.

### Constraints:

- $2 \leq \text{nums.length} \leq 5 \times 10^4$
- $0 \leq \text{nums}[i] \leq 5 \times 10^4$

## 962. Maximum Width Ramp

```
/*
    Stack
    Time complexity: O(n)
    Space complexity: O(n)
*/
class Solution {
public:
    int maxWidthRamp(std::vector<int>& nums) {
        int n=nums.size();

        std::stack<int> st;
        for (int i=0;i<n;++i) {
            if (st.empty() || nums[st.top()]>nums[i]) st.push(i);
        }

        int ans=0;
        for (int i=n-1;!st.empty()&&i>=0;--i){
            while (!st.empty() && nums[st.top()]<=nums[i]){
                ans = std::max(ans,i-st.top());
                st.pop();
            }
        }

        return ans;
    }
};
```

## 962. Maximum Width Ramp

```
class Solution {
public:
    int maxWidthRamp(std::vector<int>& nums){
        int n=nums.size();

        std::vector<int> max_suff(n);
        max_suff[n-1]=nums[n-1];
        for(int i=n-2;i>=0;--i)
            max_suff[i]=std::max(max_suff[i+1],nums[i]);

        int ans=0,i=0,j=0;
        while(i<n&& j<n){
            while(j<n && nums[i]<=max_suff[j]) j++;

            ans=std::max(ans,j-i-1);

            i++;
        }
        return ans;
    }
};
```