**Maximum of minimum for every window size**

#include <bits/stdc++.h>

vector<int> maxMinWindow(vector<int> nums, int n) {

stack<int> stk;

vector<int> left(n), right(n), ans(n, INT\_MIN);

left[0] = -1;

stk.push(0);

for (int i = 1; i < n; i++) {

while (!stk.empty() && nums[stk.top()] >= nums[i]) {

stk.pop();

}

left[i] = stk.empty() ? -1 : stk.top();

stk.push(i);

}

while (!stk.empty()) {

stk.pop();

}

right[n - 1] = n;

stk.push(n - 1);

for (int i = n - 2; i >= 0; i--) {

while (!stk.empty() && nums[stk.top()] >= nums[i]) {

stk.pop();

}

right[i] = stk.empty() ? n : stk.top();

stk.push(i);

}

for (int i = 0; i < n; i++) {

int len = right[i] - left[i] - 1;

ans[len - 1] = max(ans[len - 1], nums[i]);

}

for (int i = n - 2; i >= 0; i--) {

ans[i] = max(ans[i], ans[i + 1]);

}

return ans;

}