Two Pointers

- Priyansh Agarwal

Problem 1

Distance between two coordinates **x and y** is defined as absolute difference between the two**.**

Given an array **nums** having **n** positive integers $A_1, A_2, ..., A_n$, and an positive integer **k.**

Return the **k**th smallest distance among all the pairs of integers **nums[i]** and **nums[j]** where 0 <= i < j <n.

Constraints: $1 <= n <= 10^5$, $1 <= A_i <= 10^5$.

Sliding Window

- Useful for array based problems - subarray
- When to use?
- Optimization Technique
- Use of 2 pointers.
- Super useful for interviews too

Given an array, what is the maximum sum of a subarray of size k

Given an array, find the first negative number in every subarray of size k

Given an array, find the median of each subarray of size k

Given an array, find the minimum number in each subarray of size k

Solution:

- Sliding window
- Use of deque

```
vector<int> maxSlidingWindow(vector<int>& nums, int k) {
    deque<int> d;
    vector<int> ret;
    for(int i = 0; i < k; i++){
        while(!d.empty() && nums[i] > nums[d.back()]){
            d.pop_back();
        }
        d.push_back(i);
    }
    for(int i = k; i < nums.size(); i++){
        ret.push_back(nums[d.front()]);
        if(!d.empty() && d.front() <= i-k){
            d.pop_front();
        }
        while(!d.empty() && nums[i] >= nums[d.back()]){
            d.pop_back();
        }
        d.push_back(i);
    }
    ret.push_back(nums[d.front()]);
    return ret;
}
```