## Day 188/180 Heap Problems

### 1: Kth largest element in a stream

```cpp
class Solution {
public:
    vector<int> kthLargest(int k, int arr[], int n) {
        priority_queue<int> pq; // Create a max heap (priority queue) to store the k largest elements
        vector<int> ans(n, -1); // Initialize a vector to store the kth largest elements, initially filled with -1

        // Iterate through the array 'arr'
        for(int i = 0; i < n; i++){
            pq.push(-arr[i]); // Push the negation of each element into the max heap to simulate a min heap
            if(pq.size() > k) pq.pop(); // If the size of the max heap exceeds k, remove the smallest element

            // If the size of the max heap is equal to k, update the kth largest element in 'ans'
            if(pq.size() == k){
                ans[i] = -pq.top(); // Store the negation of the top element (kth largest) in 'ans'
            }
        }

        return ans; // Return the vector containing the kth largest elements
    }
};
```