📘 Sorting a K-Sorted Array using Min-Heap (Priority Queue)

🧠 What is a K-Sorted Array?

A **k-sorted array** is an array where each element is **at most** `k` **positions away** from its correct position in the sorted array.

📌 Example:

Original: `[10, 9, 8, 7, 4, 70, 60, 50]` , `k = 4`

Each element is at most 4 positions away from its actual sorted position.

✅ Problem Statement:

> Given a k-sorted array, use a min-heap to sort the array efficiently.

🛠️ Our Code:

```
#include<iostream>
#include<vector>
#include<queue>
using namespace std;

int main(){
  vector<int> arr = {10,9,8,7,4,70,60,50};
  int k = 4;
  vector<int> ans;

  priority_queue<int, vector<int>, greater<int>> pq; // Min-heap
  for(int i = 0; i < arr.size(); i++){
    pq.push(arr[i]);
    if(pq.size() > k){
      ans.push_back(pq.top());
      pq.pop();
    }
  }

  while(!pq.empty()){
    ans.push_back(pq.top());
    pq.pop();
  }

  for(int i = 0; i < ans.size(); i++){
    cout << ans[i] << " ";
  }

  return 0;
}
```

🔍 Logic + Dry Run:

🔄 Step-by-Step Dry Run:

Input: `arr = [10, 9, 8, 7, 4, 70, 60, 50]`, `k = 4`

- Push first `k+1` elements → min-heap will sort these internally.
- For every next element:
    - Insert it in the heap.
    - Pop and place the smallest in result (heap ensures it's the correct one).

💥 Working:

i	pq content (heap)	ans (output)
0	[10]	-

1	[9,10]	-
2	[8,10,9]	-
3	[7,8,9,10]	-
4	[4,7,9,10,8] → pop(4)	[4]
5	[7,8,9,10,70] → pop(7)	[4,7]
6	[8,10,9,70,60] → pop(8)	[4,7,8]
7	[9,10,60,70,50] → pop(9)	[4,7,8,9]

Final pop: Remaining: `10, 50, 60, 70` → sorted.

⏱️ Time & Space Complexity:

Metric	Value	Why
Time Complexity	`O(n log k)`	For each of `n` elements, heap ops take `log k`
Space Complexity	`O(k)`	Heap stores at most `k+1` elements at a time

✅ Summary:

Feature	Value
Array Type	K-sorted (each element max `k` off)
Goal	Fully sort it
Data Structure Used	Min-Heap (priority_queue)
Core Idea	Maintain heap of size `k+1`
Time Complexity	`O(n log k)`
Space Complexity	`O(k)`
Real-life Application	Streaming data, file merge, etc.

✨ Notes for Interviews:

- **Heap sort** ka real-world variation hai ye.
- Jab bhi problem bole **“nearly sorted”** ya **“element k distance pe hai”**, tab heap lagao bhai!

- Min-heap = Best jab smallest pe focus ho.

📌 Bonus Tip:

> “Jitna k ka gap, utna chhota heap — aur utna hi sorted output!”
