class Solution {

public:

vector<int> topKFrequent(vector<int>& nums, int k) {

unordered\_map<int, int> freq;

// Count frequency of each number

for (int num : nums) {

freq[num]++;

}

// Min-Heap: stores pairs {frequency, number}

priority\_queue<pair<int, int>, vector<pair<int, int>>, greater<pair<int, int>>> minHeap;

// Maintain a heap of size k

for (auto& [num, count] : freq) {

minHeap.push({count, num});

if (minHeap.size() > k) {

minHeap.pop(); // Remove least frequent element

}

}

// Extract k most frequent elements

vector<int> result;

while (!minHeap.empty()) {

result.push\_back(minHeap.top().second);

minHeap.pop();

}

return result;

}

};