**Running Median**

#include <bits/stdc++.h>

void findMedian(int\* arr, int n)

{

std::priority\_queue<int> maxHeap;

std::priority\_queue<int, std::vector<int>, std::greater<int>> minHeap;

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

if (maxHeap.empty())

maxHeap.push(arr[i]);

else if (arr[i] > maxHeap.top())

minHeap.push(arr[i]);

else

maxHeap.push(arr[i]);

if (maxHeap.size() - minHeap.size() == 2) {

minHeap.push(maxHeap.top());

maxHeap.pop();

}

else if (minHeap.size() - maxHeap.size() == 1) {

maxHeap.push(minHeap.top());

minHeap.pop();

}

if (maxHeap.size() == minHeap.size())

std::cout << (maxHeap.top() + minHeap.top()) / 2 << " ";

else

std::cout << maxHeap.top() << " ";

}

}