Code for heapify

#include <bits/stdc++.h>

using namespace std;

void heap\_sort(vector<int>& arr);

void build\_heap(vector<int>& arr);

void heapify(vector<int>& arr, int i, int heap\_size);

void heap\_sort(vector<int>& arr){

    // Build a heap from the input array

    build\_heap(arr);

    // Repeat until the heap contains only one element

    for (int i = arr.size() - 1; i > 0; i--){

        // Swap the root element with the last element

        swap(arr[0], arr[i]);

        // Remove the last element (which is now in the correct position)

        int heap\_size = i;

        heapify(arr, 0, heap\_size);

    }

    // Reverse the sorted array and return it

    reverse(arr.begin(), arr.end());

}

// building heap

void build\_heap(vector<int>& arr){

    // Build a max heap from the input array

    int n = arr.size();

    for (int i = n / 2 - 1; i >= 0; i--){

        heapify(arr, i, n);

    }

}

// function for heapify

void heapify(vector<int>& arr, int i, int heap\_size){

    // Heapify the subtree rooted at i in the input array

    int largest = i;

    int left = 2 \* i + 1;

    int right = 2 \* i + 2;

    if (left < heap\_size && arr[left] > arr[largest])

        largest = left;

    if (right < heap\_size && arr[right] > arr[largest])

        largest = right;

    if (largest != i){

        swap(arr[i], arr[largest]);

        heapify(arr, largest, heap\_size);

    }

}