Heap

#include <iostream>

#include <vector>

using namespace std;

void swap(int\* a, int\* b)

{

int temp = \*b;

\*b = \*a;

\*a = temp;

}

void heapify(vector<int>& heap,int size, int i)

{

int largest = i;

int l = (2 \* i )+ 1;

int r = (2 \* i) + 2;

if (l < size && heap[l] > heap[largest])

largest = l;

if (r < size && heap[r] > heap[largest])

largest = r;

if (largest != i)

{

swap(&heap[i], &heap[largest]);

heapify(heap, size,largest);

}

}

void insert(vector<int>& heap, int element)

{

heap.push\_back(element);

int size = heap.size();

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

{

heapify(heap, size,i);

}

}

//for sorting heap and to make priority queue

void heapSort(vector<int>& heap) {

int size = heap.size();

for (int i = size / 2 - 1; i >= 0; i--) //heapifying heap if necessary

heapify(heap, size, i);

for (int i = size - 1; i >= 0; i--) { //sorting heap

swap(&heap[0], &heap[i]);

heapify(heap, i, 0);

}

int counter = 1;

for (int i = size - 1; i >= 0; i--) { //defining priorites

cout << "Priority " << counter << " -> " << heap[i] << "\n";

counter += 1;

}

}

void printArray(vector<int>& heap)

{

for (int i = 0; i < heap.size(); ++i)

cout << heap[i] << " ";

cout << "\n";

}

int main()

{

vector<int> heap;

insert(heap, 10);

insert(heap, 20);

insert(heap, 15);

insert(heap,30);

insert(heap, 40);

cout << "Max-Heap array: ";

printArray(heap);

heapSort(heap);

//cout << "Sorted Heap : ";

//printArray(heap);

}