**Max-heap with deletion**

#include <iostream>

using namespace std;

void maxHeapify(int arr[], int n, int i) {

int largest = i;

int left = 2 \* i + 1;

int right = 2 \* i + 2;

if (left < n && arr[left] > arr[largest]){

largest = left;

}

if (right < n && arr[right] > arr[largest]){

largest = right;

}

if (largest != i) {

swap(arr[i], arr[largest]);

maxHeapify(arr, n, largest);

}

}

void buildMaxHeap(int arr[], int n) {

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

maxHeapify(arr, n, i);

}

}

int deleteMax(int arr[], int &n) {

if (n <= 0) {

cout << "Heap is empty!";

return -1;

}

int root = arr[0];

arr[0] = arr[n-1];

n--;

maxHeapify(arr, n, 0);

return root;

}

int main() {

int n;

cout<<"Enter no. of elements: ";

cin>>n;

int arr[n];

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

cout<<"Enter arr["<<i<<"]: ";

cin>>arr[i];

}

cout<<"\nOriginal Array: ";

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

cout<<arr[i]<<" ";

}

cout << endl;

buildMaxHeap(arr, n);

cout<<"\nMax Heap: ";

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

cout<<arr[i]<<" ";

}

cout<<endl;

int delitem = deleteMax(arr, n);

if (delitem != -1) {

cout<<"\nMaximum element deleted: "<<delitem<<endl;

}

cout << "\nMax Heap after deletion: ";

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

cout<<arr[i]<<" ";

}

cout << endl;

return 0;

}

**Min-heap with deletion**

#include <iostream>

using namespace std;

void minHeapify(int arr[], int n, int i) {

int smallest = i;

int left = 2 \* i + 1;

int right = 2 \* i + 2;

if (left < n && arr[left] < arr[smallest]){

smallest = left;

}

if (right < n && arr[right] < arr[smallest]){

smallest = right;

}

if (smallest != i) {

swap(arr[i], arr[smallest]);

minHeapify(arr, n, smallest);

}

}

void buildMinHeap(int arr[], int n) {

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

minHeapify(arr, n, i);

}

}

int deleteMin(int arr[], int &n) {

if (n <= 0) {

cout << "Heap is empty!";

return -1;

}

int root = arr[0];

arr[0] = arr[n-1];

n--;

minHeapify(arr, n, 0);

return root;

}

int main() {

int n;

cout<<"Enter no. of elements: ";

cin>>n;

int arr[n];

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

cout<<"Enter arr["<<i<<"]: ";

cin>>arr[i];

}

cout<<"\nOriginal Array: ";

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

cout<<arr[i]<<" ";

}

cout << endl;

buildMinHeap(arr, n);

cout<<"\nMin Heap: ";

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

cout<<arr[i]<<" ";

}

cout<<endl;

int delitem = deleteMin(arr, n);

if (delitem != -1) {

cout<<"\nMinimum element deleted: "<<delitem<<endl;

}

cout << "\nMin Heap after deletion: ";

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

cout<<arr[i]<<" ";

}

cout<<endl;

return 0;

}

**Heap-sort using Max-heap**

#include <iostream>

using namespace std;

void maxHeapify(int arr[], int n, int i) {

int largest = i;

int left = 2 \* i + 1;

int right = 2 \* i + 2;

if (left < n && arr[left] > arr[largest]){

largest = left;

}

if (right < n && arr[right] > arr[largest]){

largest = right;

}

if (largest != i) {

swap(arr[i], arr[largest]);

maxHeapify(arr, n, largest);

}

}

void buildMaxHeap(int arr[], int n) {

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

maxHeapify(arr, n, i);

}

}

void heapSort(int arr[], int n) {

buildMaxHeap(arr, n);

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

swap(arr[0], arr[i]);

maxHeapify(arr, i, 0);

}

}

int main() {

int n;

cout<<"Enter no. of elements: ";

cin>>n;

int arr[n];

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

cout<<"Enter arr["<<i<<"]: ";

cin>>arr[i];

}

cout<<"\nOriginal Array: ";

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

cout<<arr[i]<<" ";

}

cout<<endl;

heapSort(arr, n);

cout<<"\nSorted array: ";

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

cout<<arr[i]<<" ";

}

cout<<endl;

return 0;

}

**Heap-sort using Min-heap**

#include <iostream>

using namespace std;

void minHeapify(int arr[], int n, int i) {

int smallest = i;

int left = 2 \* i + 1;

int right = 2 \* i + 2;

if (left < n && arr[left] < arr[smallest]){

smallest = left;

}

if (right < n && arr[right] < arr[smallest]){

smallest = right;

}

if (smallest != i) {

swap(arr[i], arr[smallest]);

minHeapify(arr, n, smallest);

}

}

void buildMinHeap(int arr[], int n) {

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

minHeapify(arr, n, i);

}

}

void heapSort(int arr[], int n) {

buildMinHeap(arr, n);

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

swap(arr[0], arr[i]);

minHeapify(arr, i, 0);

}

}

int main() {

int n;

cout<<"Enter no. of elements: ";

cin>>n;

int arr[n];

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

cout<<"Enter arr["<<i<<"]: ";

cin>>arr[i];

}

cout<<"\nOriginal Array: ";

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

cout<<arr[i]<<" ";

}

cout<<endl;

heapSort(arr, n);

cout<<"\nSorted array: ";

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

cout << arr[i] << " ";

}

cout << endl;

return 0;

}