

# DSA Assignment 9

Name: Jashnoor Singh

Roll: 1024030106

Q1:

```
// implement heap sort in ascending order

#include <iostream>
using namespace std;

void heapify(int arr[], int size, int i){
    int largest = i;
    int left = 2*i + 1;
    int right = 2*i + 2;
    if(left<size && arr[largest]<arr[left])largest = left;
    if(right<size && arr[largest]<arr[right])largest = right;
    if(largest != i){
        swap(arr[i],arr[largest]);
        heapify(arr,size,largest);
    }
}

void heap_sort(int arr[], int size){
    if(size == 0)return;
    swap(arr[size-1],arr[0]);
    size--;
    heapify(arr,size,0);
    heap_sort(arr,size);
}

int main(){
    int arr[] = {37, 12, 89, 45, 23, 71, 6, 54, 18, 90};
    for(int i=(10/2) - 1;i>=0;i--) { // used to heapify the array
        heapify(arr,10,i);
    }
    for(int i=0;i<10;i++){
        cout<<arr[i]<<' ';
    }
    cout<<endl;
    heap_sort(arr,10); // used to sort in ascending order of heap
    for(int i=0;i<10;i++){
        cout<<arr[i]<<' ';
    }
}
```

Output:

```
Array after max heapification
90 54 89 45 23 71 6 37 18 12
Final arry after sorting
6 12 18 23 37 45 54 71 89 90 %
```

Q2:

```
// implement heap sort in descending order

#include <iostream>
using namespace std;
void heapify(int arr[], int size, int i){
    int smallest = i;
    int left = 2*i + 1;
    int right = 2*i + 2;
    if(left<size && arr[smallest]>arr[left])smallest = left;
    if(right<size && arr[smallest]>arr[right])smallest = right;
    if(smallest != i){
        swap(arr[i],arr[smallest]);
        heapify(arr,size,smallest);
    }
}
void heap_sort(int arr[], int size){
    if(size == 0) return;
    swap(arr[size-1],arr[0]);
    size--;
    heapify(arr,size,0);
    heap_sort(arr,size);
}
int main(){
    int arr[] = {37, 12, 89, 45, 23, 71, 6, 54, 18, 90};
    for(int i=(10/2)-1;i>=0;i--) { // array heapified to min heap
        heapify(arr,10,i);
    }
    for(int i=0;i<10;i++){
        cout<<arr[i]<<' ';
    }
    heap_sort(arr,10);
    cout<<endl;
    for(int i=0;i<10;i++){
        cout<<arr[i]<<' ';
    }
}
```

Output:

```
Array after min heapification
6 12 37 18 23 71 89 54 45 90
Array after sorting
90 89 71 54 45 37 23 18 12 6 %
```

Q3:

```
// implement priority queue using heaps

#include <iostream>
using namespace std;

class heap{
public:
    int arr[100];
    int size = 0;
    void insert(int value){
        size = size+1;
        int index = size;
        arr[index] = value;
        while(index>1){
            int par_index = index/2;
            if(arr[par_index]<arr[index]){
                swap(arr[par_index],arr[index]);
                index = par_index;
            }
            else{
                return;
            }
        }
    }
    void deletion(){
        if(size==0){
            cout<<"Nothing to delete";
            return;
        }
        else{
            arr[1] = arr[size];
            size--;
            int i=1;
            while(i<size){
                int left_child = 2*i;
                int right_child = 2*i + 1;
                if(left_child<=size && arr[left_child]>arr[i]){
                    swap(arr[i],arr[left_child]);
                    i=left_child;
                }
                else if(right_child<=size && arr[right_child]>arr[i]){
                    swap(arr[i],arr[right_child]);
                    i=right_child;
                }
                else{ // at correct place
                    return;
                }
            }
        }
    }
}
```

```
        }
    }
    int top(){
        return arr[1];
    }
};

int main(){
    heap h;
    h.insert(10);
    h.insert(20);
    h.insert(1);
    h.insert(11);
    h.insert(5);
    cout<<"Top is "<<h.top()<<endl;
    h.deletion();
    cout<<"New top is "<<h.top();
}
```

Output:

```
Top is 20
New top is 11%
```