

ASSIGNMENT: 6

AIM Read the marks obtained by the students of second year in an online examination of a particular subject. Find out maximum and minimum marks obtained in that subject using heap data structure.

OBJECTIVE: To study and learn the concepts of heap data structure.

THEORY: Heap definition- It is a Complete (Binary) Tree with each node having HEAP PROPERTY. Elements are filled level by level from left- to-right. If A is a parent node of B, then the key (the value) of node A is ordered with respect to the key of node B with the same ordering applying across the heap.

Types of heap: 1) Min heap

2) Max heap

○ **MAX HEAP definition:**

Complete (Binary) tree with the property that the **value of each node** is at least as large as the value of its children (i.e. \geq value of its children)

○ **MIN HEAP definition:**

Complete (Binary) tree with the property that the **value of each node** is at most as large as the value of its children (i.e. \leq value of its children)

ALGORITHM To maintain the max heap property i.e. MAXHEAPIFY

MAX- HEAPIFY(A, i, n)

1. $l \leftarrow \text{LEFT}(i)$
2. $r \leftarrow \text{RIGHT}(i)$
3. **if** $l \leq n$ and $A[l] > A[i]$
4. **then** largest $\leftarrow l$
5. **else** largest $\leftarrow i$
6. **if** $r \leq n$ and $A[r] > A[\text{largest}]$
7. **then** largest $\leftarrow r$

8. **if** largest \neq i
9. **then** exchange A[i] \leftrightarrow A[largest]
10. MAX- HEAPIFY(A, largest, n)

PROGRAM

```
#include<iostream>
using namespace std;
class heap
{
public:
void print array(int a[], int n);
void heapsort(int a[], int n);
void minimum(int a[],int n);
void maximum(int a[],int n);
};
void heapify(int a[],int n,int i);
void heap:: heapsort(int a[], int n)
{
for(int i=(n/2)- 1; i>=0;i-- )
{
heapify(a,n,i);
}
for(int i=(n- 1);i>=0;i-- )
{
int temp= a[0];
a[0]= a[i];
a[i]= temp;
heapify (a,i,0);
}
}
void heapify(int a[],int n, int i)
{
int largest=i;
int l= (2*i)+1;
int r=(2*i)+2;
if(l<n && a[l]>a[largest])
largest=l;
if(r<n && a[r]>a[largest])
largest=r;

if(largest!=i)
{
int t = a[i];
a[i]=a[largest];
a[largest]=t;
heapify(a,n,largest);
}
}
```

```

void heap:: print array(int a[],int n)
{
    for(int i=0;i<n;i++)
    {
        cout<<a[i]<<" ";
        cout<<"\n";
    }
}

void heap::maximum(int a[],int n)
{
    cout<<"MAXI MUM MARKS:"<<a[n- 1]<<endl;
}

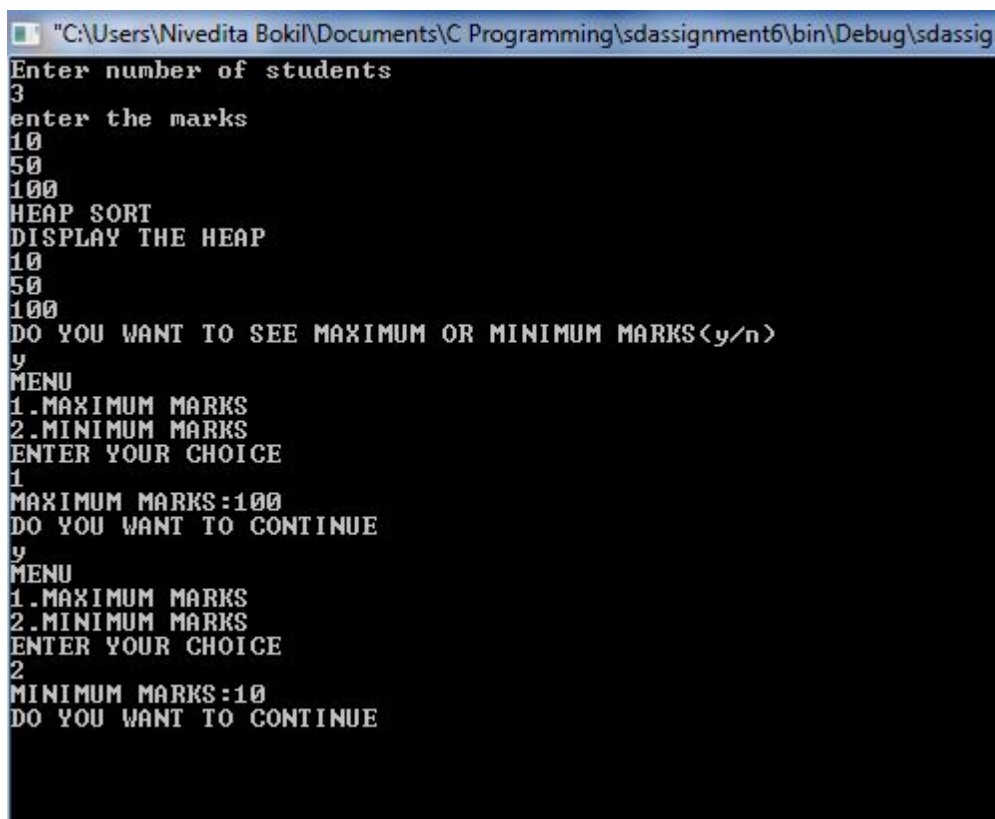
void heap::minimum(int a[],int n)
{
    cout<<"MI NI MUM MARKS:"<<a[0]<<endl;
}

int main()
{
    heap h;
    int a[100],n;
    cout<<"Enter number of student s"<<endl;
    cin>>n;
    cout<<"enter the marks"<<endl;
    for(int i=0;i<n;i++)
    {
        cin>>a[i];
    }
    cout<<"HEAP SORT"<<endl;
    h.heapsort(a,n);
    cout<<"DI SPLAY THE HEAP"<<endl;
    h.print array(a,n);
    char ch;
    int choice;
    cout<<"DO YOU WANT TO SEE MAXI MUM OR MI NI MUM MARKS(y/n)"<<endl;
    cin>>ch;
    while(ch=='y')
    {
        cout<<"MENU"<<endl;
        cout<<"1. MAXI MUM MARKS"<<endl;
        cout<<"2. MI NI MUM MARKS"<<endl;
        cout<<"ENTER YOUR CHOI CE"<<endl;
        cin>>choice;
        swit ch(choice)
        {
            case 1:
                h.maximum(a,n);
                break;
            case 2:
                h.minimum(a,n);
                break;
            default:

```

```
        cout << "SORRY! WRONG CHOICE" << endl;
        break;
    }
    cout << "DO YOU WANT TO CONTINUE" << endl;
    cin >> ch;
}
return 0;
}
```

OUTPUT:



```
"C:\Users\Nivedita Bokil\Documents\C Programming\sdassignment6\bin\Debug\sdassig
Enter number of students
3
enter the marks
10
50
100
HEAP SORT
DISPLAY THE HEAP
10
50
100
DO YOU WANT TO SEE MAXIMUM OR MINIMUM MARKS(y/n)
y
MENU
1.MAXIMUM MARKS
2.MINIMUM MARKS
ENTER YOUR CHOICE
1
MAXIMUM MARKS:100
DO YOU WANT TO CONTINUE
y
MENU
1.MAXIMUM MARKS
2.MINIMUM MARKS
ENTER YOUR CHOICE
2
MINIMUM MARKS:10
DO YOU WANT TO CONTINUE
```

CONCLUSION: We successfully implemented heap data structure.

