DAA Lab-9

Name: Kshitij Kumar Sharma Roll No.: 1905514 Date: 01/10/2021

**Q-8.1)** Write a menu-based program to implement the priority queue operations.

User Choice:

1 (to insert an element)

2 (to get maximum)

3 (to extract maximum)

4 (to increase the specified key to a given value)

5 (print order-level traversal of queue), and

6 (to exit).

In the beginning as well as after every operation you are supposed to print the options 1-6 to get an (user) input.

**Program:**

/\*

Written by: Kshitij Kumar Sharma Roll No.: 1905514

Idea of the solution:

INSERT(S,x): inserts the element x into the set S, which is equivalent to the operation S = S x ∪ MAXIMUM(S): returns the element of S with the largest key EXTRACT-MAX(S): removes and returns the element of S with the largest key INCREASE-KEY(S,x,k): increases the value of element x’s key to the new value k, which is assumed to be at least as large as x’s current key value

\*/

#include<bits/stdc++.h>

using namespace std;

void heapMax(int a[]) //prints maximum heap element

{

    cout<<a[0];

}

void heapify(int a[], int n, int i) //make max heap

{

    int l,r,largest=i;

    l=2\*i+1;

    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 heapExtractMax(int a[],int& n) //deletes the max element

{

    int l=a[n-1];

   a[0]=l;

   n=n-1;

   heapify(a,n,0);

}

void heapIncreaseKey(int a[],int n,int i) //increases the key value

{

     int p=(i-1)/2;

    if(a[p]>0)

    {

        if(a[i]>a[p])

        {

            swap(a[i],a[p]);

            heapIncreaseKey(a,n,p);

        }

    }

}

void maxHeapInsert(int a[],int& n, int k)

{

    n=n+1;

    a[n-1]=k;

    heapIncreaseKey(a,n,n-1);

}

void print(int a[],int n)

{

    int i;

    for(i=0;i<n;i++)

        cout<<a[i]<<" ";

}

void increaseKey(int a[],int i,int k)

{

    if(k<a[i])

        {

            cout<<"Error \n";

            return;

        }

    a[i]=k;

    while(i>1 && a[(i-1)/2]<a[i])

    {

        swap(a[i],a[(i-1)/2]);

        i=(i-1)/2;

    }

}

int main() //driver code

{

    int n;

    cout<<"Enter size : ";

    cin>>n;

    int i,a[n],ch;

    cout<<"Enter elements : \n";

    for(i=0;i<n;i++)

        cin>>a[i];

    sort(a,a+n,greater<int>());

    do

    {

        cout<<"\n";

        cout<<"MENU \n";

        cout<<"1-Insert       \t\t 2-Maximum \t\t 3-Extract maximum \n";

        cout<<"4-Increase key \t\t 5-Print   \t\t 6-Exit \n";

        cout<<"Enter the choice : ";

        cin>>ch;

        if(ch==1)

        {

            int k;

            cout<<"Enter key to insert : ";

            cin>>k;

            maxHeapInsert(a,n,k);

        }

        if(ch==2)

            heapMax(a);

        if(ch==3)

            heapExtractMax(a,n);

        if(ch==4)

        {

            int k,j;

            cout<<"Enter index : ";

            cin>>j;

            cout<<"Enter key : ";

            cin>>k;

            increaseKey(a,j,k);

        }

        if(ch==5)

            print(a,n);

    }while(ch<6);

}

**Output:**



