# Heap sort with time analysis

# Code:

#include<iostream>

#include<math.h>

using namespace std;

int count=0;

void heapify(int arr[],int n,int i){

    int lar = i;

    int l = 2\*i;

    int r = 2\*i+1;

    count++;

    if(l<n && arr[l]>arr[lar]){

        lar = l;

    }

    count++;

    if(r<n && arr[r]>arr[lar]){

        lar = r;

    }

    // count++;

    if(lar!=i){

        swap(arr[i],arr[lar]);

        heapify(arr,n,lar);

    }

}

void heapsort(int arr[],int n){

    for(int i=n/2;i>=1;i--){

         count++;

        heapify(arr,n,i);

    }

    for(int i=n-1;i>1;i--){

        swap(arr[1],arr[i]);

        heapify(arr,i,1);

    }

}

int main(){

    int arr[1000],desc[1000],ran[1000];

    int n;

    cout<<"Enter size of array : \n";

    cin>>n;

    // cout<<"enter "<<n<<" numbers :\n";

    // for(int i=1;i<=n;i++){

    //     cin>>arr[i];

    // }

    // cout<<"Entered array :\n";

    // for(int i=1;i<=n;i++){

    //     cout<<arr[i]<<" ";

    // }

    // for(int i=1;i<=n;i++){

    //    arr[i] = i;

    // }

    // for(int i=1;i<=n;i++){

    //    desc[i] = n-i;

    //  }

    for(int i=1;i<=n;i++){

       ran[i] = rand();

     }

    //  heapsort(arr,n+1);

    // //  cout<<"\nascending\n";

    //  cout<<"\nsorted array :\n";

    // for(int i=1;i<=n;i++){

    //     cout<<arr[i]<<" ";

    // }

    //  heapsort(desc,n+1);

    // cout<<"\ndescending\n";

    // cout<<"\nsorted array :\n";

    // for(int i=1;i<=n;i++){

    //     cout<<desc[i]<<" ";

    // }

     heapsort(ran,n+1);

    //  cout<<"\nRandom\n";

    // cout<<"\nsorted array :\n";

    // for(int i=1;i<=n;i++){

    //     cout<<ran[i]<<" ";

    // }

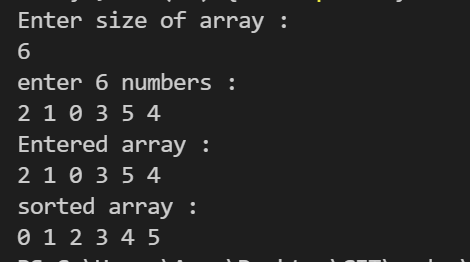
    cout<<"\ncount : "<<count<<endl;

    cout<<"nlogn : "<<count/(n\*log(n))<<"\nn^2 : "<<count/float(n\*n);

return 0;

}

## Output:



## Analysis:

