**Name:** Purvi Singhvi

**Enroll. No.:** 180099

**PROGRAM- 9**

**Aim:** Write an algorithm and program to implement Heap sort.

**Algorithm:**

**1.** Build a max heap from the input data.  
**2.** At this point, the largest item is stored at the root of the heap. Replace it with the last item of the heap followed by reducing the size of heap by 1. Finally, heapify the root of the tree.  
**3.** Repeat step 2 while size of heap is greater than 1.

**Source code:**

#include<stdio.h>

#include<conio.h>

void heapify(int arr[], int n, int i)

{

int largest = i;

int l = 2\*i + 1;

int r = 2\*i + 2;

if (l < n && arr[l] > arr[largest])

largest = l;

if (r < n && arr[r] > arr[largest])

largest = r;

if (largest != i)

{

int t;

t=arr[i];

arr[i]=arr[largest];

arr[largest]=t;

heapify(arr, n, largest);

}

}

void heapSort(int arr[], int n)

{

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

heapify(arr, n, i);

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

{

int t;

t=arr[0];

arr[0]=arr[i];

arr[i]=t;

heapify(arr, i, 0);

}

}

void printArray(int arr[], int n)

{

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

printf("%d ", arr[i]);

printf("\n");

}

void main()

{

int j,k;

int arr[10];

int n,i;

clrscr();

printf("Enter the size of array:");

scanf("%d",&n);

printf("Enter the elements of an array:");

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

{

scanf("%d",&arr[i]);

}

heapSort(arr, n);

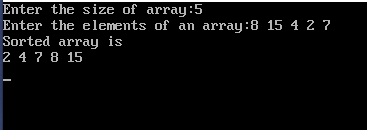
printf( "Sorted array is \n");

printArray(arr, n);

getch();

}

**Output:**



**Complexity:**

Worst case: O(n log(n))

Average case: O(n log(n))

Best case: O(n log(n))