ALGORITHM

1. Start
2. swap(*a,*b)
    2.1. Declare temp=a
    2.2. Set a=b
    2.3. Set b=temp
3. heapify(a[], size, i)
    3.1. Declare lc=2*i
    3.2. Declare rc=(2*i)+1
    3.3. Declare large=i
    3.4. Check if(lc<=size and a[lc]>a[large])
        3.4.1. Set large=lc
    3.5. Check if(rc<=size and a[rc]>a[large])
        3.5.1. Set large=rc
    3.6. Check if(large!=i)
        3.6.1. swap(&a[i], &a[large])
        3.6.2. heapify(a,size,large)
4. End function
5. buildheap(a[], n)
    5.1. for i=n/2, i>=1 do
        5.1.1. heapify(a, n, i)
6. End function
7. heapsort(a[], n)
    7.1. buildheap(a,n)
    7.2. for i=n, i>=1 do
        7.2.1. swap(&a[1], &a[i])
        7.2.2. heapify(a,i-1,1)
8. End function
9. main()
    9.1. Declare size
    9.2. Print, Enter the number of elements.
    9.3. Read the size.

Date: 30/10/24

PROGRAM NO: 20
HEAP SORT

Aim: To implement heap sort.

PROGRAM

```
#include<stdio.h>

void swap(int *a,int *b)
{
   int temp=*a;
   *a=*b;
   *b=temp;
}

void heapify(int a[], int size, int i)
{
   int lc=2*i;
   int rc=(2*i)+1;
   int large=i;
   if(lc<=size && a[lc]>a[large])
      large=lc;
   if(rc<=size && a[rc]>a[large])
      large=rc;
   if(large!=i)
   {
         swap(&a[i], &a[large]);
         heapify(a,size,large);
   }
}

void buildheap(int a[], int n)
{
   for(int i=n/2;i>=1;i--)
      heapify(a, n, i);
}

void heapsort(int a[], int n)
{
   buildheap(a,n);
   for(int i=n;i>=1;i--)
   {
      swap(&a[1], &a[i]);
      heapify(a,i-1,1);
   }
}

void main()
{
   int size;
    printf("Enter the number of elements:\n");
   scanf("%d",&size);
```

9.4. Declare a[size+1]

9.5. Print, Enter elements.

9.6. for i=1, i<=size do

9.6.1. Read the elements.

9.7. heapsort(a, size)

9.8. for i=1, i<=size do

9.8.1. Print, a[i]

10. End function

11. Stop

Output

Enter the number of elements: 5

Enter 5 elements: 6 9 2 5 4

2 4 5 6 9

```
    int a[size+1];
    printf("Enter %d elements\n", size);
    for(int i=1;i<=size;i++)
        scanf("%d", &a[i]);
    heapsort(a, size);
    for(int i=1;i<=size;i++)
        printf("%d\t", a[i]);
}
```

Result:

The program is executed successfully and output is obtained.