20. To develop a program to important heap sort technique.

:Heapsort is a comparision based sorting technique based on a Binary Heap data structre. It is similar to selection sort where we first find the maximum element and place the maximum element at the end. We repeat the same process for the remaining element.

The time complexity of heapify is O(log(n))
The time complexity of create and BuildHeap() is O(n)
The overall time complexity of Heap sort is
O(n*log(n))
Auxiliary Space :- O(logn)

Code:

```
def heapify(array, a, b):
    largest = b
    l = 2*b+1
    root = 2*b+2
    if l<a and array[b] < array[l]:
        largest = l
    if root<a and array[largest] < array[root]:
        largest = root
    if largest != b:
        array[b], array[largest] = array[largest], array[b]
        heapify(array, a, largest)
def HeapSort(array):
    a = len(array)
    for b in range(a//2-1, -1, -1):
        heapify(array, a, b)
    for b in range(a-1, 0, -1):
```

```
                array[b], array[0] = array[0], array[b]
                heapify(array, b, 0)
array = [7,2,5,6,3,1,8,4]
Heap_sort(array)
a = len(array)
print("Array after sorting is:", array)
```

Output-

Array after sorting is : [1, 2, 3, 4, 5, 6, 7]