Name : Yash Srivastava || Section - A || Roll. No - 63

# Implementation and Analysis of Heap Sort

```
def heapify(arr, n, i):
   largest = i
   l = 2 * i + 1
   r = 2 * i + 2
   if l < n and arr[l] > arr[largest]:
       largest = l
   if r < n and arr[r] > arr[largest]:
       largest = r
   if largest != i:
       arr[i], arr[largest] = arr[largest], arr[i]
       heapify(arr, n, largest)


def heap_sort(arr):
   n = len(arr)
   for i in range(n // 2 - 1, -1, -1):
       heapify(arr, n, i)
   for i in range(n - 1, 0, -1):
       arr[i], arr[0] = arr[0], arr[i]
       heapify(arr, i, 0)


print("Enter an array")
a = list(map(int, input().split()))
heap_sort(a)
print("Sorted array is")
for i in range(len(a)):
   print(a[i], end=' ')
```

**Input:**

10 50 20 30 40 60

Name : Yash Srivastava || Section - A || Roll. No - 63

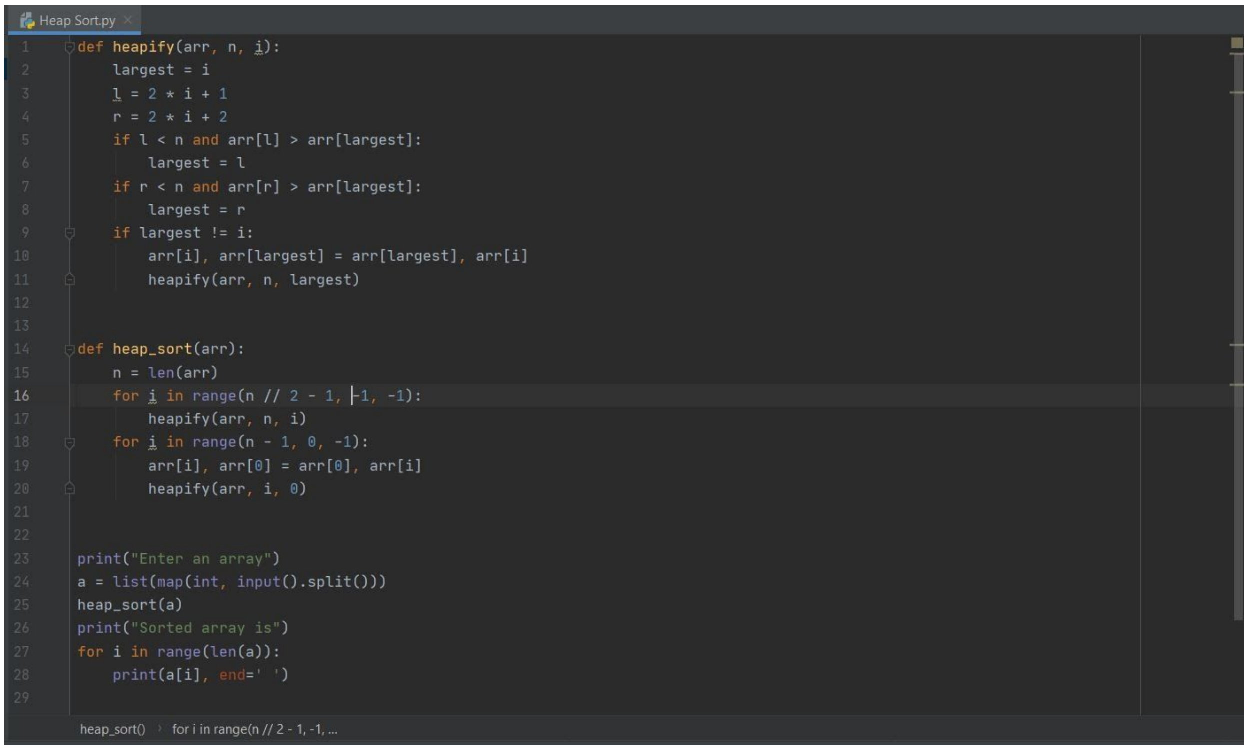
## Output:

10 20 30 40 50 60

## Time Complexity:

- O(logn)

## Program:



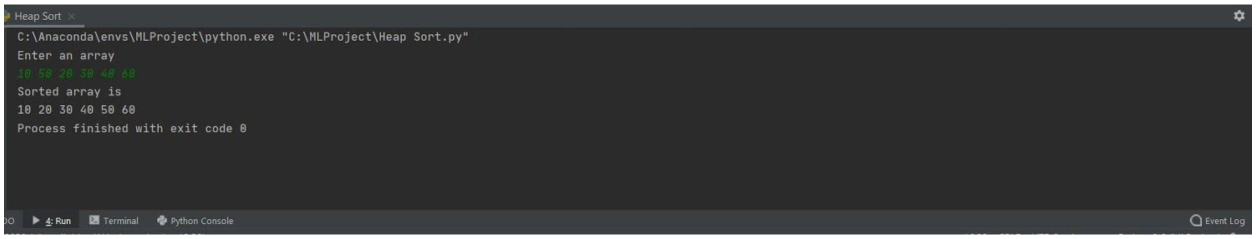
```python
def heapify(arr, n, i):
    largest = i
    l = 2 * i + 1
    r = 2 * i + 2
    if l < n and arr[l] > arr[largest]:
        largest = l
    if r < n and arr[r] > arr[largest]:
        largest = r
    if largest != i:
        arr[i], arr[largest] = arr[largest], arr[i]
        heapify(arr, n, largest)


def heap_sort(arr):
    n = len(arr)
    for i in range(n // 2 - 1, -1, -1):
        heapify(arr, n, i)
    for i in range(n - 1, 0, -1):
        arr[i], arr[0] = arr[0], arr[i]
        heapify(arr, i, 0)


print("Enter an array")
a = list(map(int, input().split()))
heap_sort(a)
print("Sorted array is")
for i in range(len(a)):
    print(a[i], end=' ')

```

## Input/Output:



```
C:\Anaconda\envs\MLProject\python.exe "C:\MLProject\Heap Sort.py"
Enter an array
10 50 20 30 40 60
Sorted array is
10 20 30 40 50 60
Process finished with exit code 0
```