Time Complexity of functions

(Here n is size of input array)

Functions in Class maxHeap:

- find_parent: O(1)
  - takes constant time
- find_lchild: O(1)
  - takes constant time
- find_rchild: O(1)
  - takes constant time
- sift_up: O(log n)
  - as this function recursively calls itself hence time complexity is O(h) or O(log n)

- sift_down: O(log n)
  - as this function recursively calls itself hence time complexity is O(h) or O(log n)

- is_valid_index: O(1)
  - takes constant time
- get_max: O(1)
  - as it just returns element of array(i.e. In constant time)
- delete_max: O(log n)
  - As it is sum of constant time and time taken by sift_down function(i.e. O(log n) + k*O(1) = O(log n))
- insert: O(log n)
  - as it calls sift_up function (O(log n) + k*O(1) = O(log n))

HeapSort Function:

Time complexity is: O(n*log n)
(as for loop run n times and each time it calls delete function)

Input and Output:

```
The Random Array is:
{ 871, 20989, 843, 1828, 10112, 10533, 2228, 22144, 6631,
12991, 17088, 14354, 2376, 24210, 11938, 3732, 32392, 16065,
20359, 25552,  }

The Output Array is:
{ 843, 871, 1828, 2228, 2376, 3732, 6631, 10112, 10533, 11938,
12991, 14354, 16065, 17088, 20359, 20989, 22144, 24210, 25552,
32392,  }
```

Output obtained is a sorted array.