Since function delMax() and function daryHeapsort() are in the worst case the Big-O complexity is 2 N Lg N

Heapsort uses <= 2 N Lg N compares and exchanges
For in-place sorting algorithm with N log N worst-case for heapsort -> yes!

For d-ary heap for order-of-growth of running time for priority queue with N items the insert() is log N, delMax() is d log$_d$N and Max() is 1

```
void insert(int k)
        {
        heap[++num] = k;
        swim(num);
        counter++;
        }
```

```
int delMax() {
        int max = heap[1];
        swap (1, num--);
        sink(1);
        heap[num+1] = 0;
        return max;
        }
```

the insert and delMax function cause the Big-O complexity
Heapsort is optimal for both time and space but it’s not stable and makes poor use of cache memory as here the insert and delMax function along with in-place sorting algorithm cause the Big-O complexity.   for in-place N log N guarantees