1. Heapsort is divided into two steps.
   1.1. Insert all the values into a heap.
   1.2. Remove all the values from the heap and store them in sequence in an array.

2. Inserting a single value on heap requires $\log_2 n+1$ operations. Recall that the number of operations required to insert a value on the heap is a function of the number of nodes in the path defined by the height, h, of the heap. $h=\lfloor \log_2 n \rfloor$ where n is the number of values stored in the heap. The number of nodes in the path from a leaf to the root is always one more than the height. Thus, the number of operations, $T(n)$, required to insert a single value is, $T(n)=cf(\log_2 n+1)$.
3. The cost of inserting all the values, n, is the product of inserting a single value n times. Since the heap grows with the insertion of each value, the cost of inserting each value is no greater than inserting the last value. $T(n)=cf(n+n\log_2 n)$.
4. The cost of removing all the values is the same as inserting all the values.
   $T(n)=cf(2n+2n\log_2 n) \Rightarrow T(n)=O(n\log_2 n)$