One swapping is done outside the while loop. Total operations for one iteration of the loop

$= 3*(i-1) +1 = 3i-2$

Since the *for* loop executes n-1 times, total number of basic operations performed

$= 3(2+3+4.....n) - 2(n-1)$

$= 3\left(\frac{n(n+1)}{2} -1\right) - 2(n-1) =$

$= = 3\left(\frac{n^2+n-2}{2}\right) - 2(n-1) = O(n^2)$

Hence the worst-case complexity of insertion sort algorithm = $O(n^2)$

3.6 Analysis of Sorting Algorithm - Heap Sort

As we have seen during the videos, the heap sort algorithm is executed in two phases. In the first phase, the list of entries is built into a max heap. In the second phase, the actual sorting is done by moving the root element, which is also the largest item, to the end of the array. Then, the size of the heap is reduced by one and the elements are adjusted to bring them to a max heap again. This process is repeated till the size of the elements comes down to one.

The algorithm is shown in Figure-14

‹ qsort() Library Function Demo (/?q=MULNCourseBook/listDetailedCBLearningContents/178099/cidb/full/view/153784/431550/431535/39731)