Homework 4 solutions

Ex 1

Best case: The list can be any sorted list of 5 elements: Example: [5, 7, 9, 11, 15

and the number to be inserted must be less than or equal to the first number in the list : Example : 3

Worst case: Again any list of 5 numbers in sorted order: Example: [4, 7, 9, 15, 17]

and the number to insert should be strictly greater than the last number in the list, so 25 would work.

Ex 2: In the worst case the algorithm performs 2n + 1 comparisons. At each step the number to be inserted is compared against the first number in the tail, and before that there is another comparison to determine whether the list is empty.

Every time the function is called with a non-empty list there are two comparisons and the list is non-empty exactly n times which gives us 2n. There is one additional comparison when the list is empty, which gives us 2n + 1

∴ complexity funct is $f(n) = 2n+1$

To show that $f(n) \in O(n)$, we must find c and n_0 such that:

$$c . n \geq 2n + 1 , \forall n > n_0$$

let $c = 3$

$$3n = 2n + n \geqslant 2n + 1, \forall n > 1$$

$\therefore$ if $c = 3$ and $n_0 = 1$, we have proved that $f(n) \in O(n)$

Ex 3.

We can immediately see that the sorting procedure calls the insert function $n - N$ times for an input of size n

$\therefore$ Insertion Sort is in $O(n^2)$