1. Derive the formulas for (i) number of comparisons, and (ii) average-case number of swaps for bubble sort.

i) number of comparisons for bubble sort

There are $n-1$ passes in the bubble sort.

Then, the first pass is $n-1$ comparissons, second pass is $n-2$ comparissons, and so on until the last pass has only 1 comparisson.

Therefore, the total number of comparissons are ⇒

$(n-1)+(n-2)+(n-3)+\ldots\ldots+1$ which is a sequence that can be written as: $\boxed{\dfrac{[n(n-1)]}{2}} \Rightarrow O(n^2)$

ii) average-case number of swaps for bubble sort

The bubble sort algorith swaps when the two adjacent element in an array are not in order. Now the occurence of swap depends on a worst case scenario (two adjacent elements are not in order) or best case scenario (two adjacent elements are in order). Hence the probability of a swap occuring is $\frac{1}{2}$. Now the total number of comparissons is $\frac{n(n-1)}{2}$ (calculated above), so the average-case number of swaps is ⇒

$$\frac{n(n-1)}{2} \times \frac{1}{2} = \boxed{\frac{n(n-1)}{4}} \Rightarrow O(n^2)$$

4. Separately plot the results of #comparisons and #swaps by input size, together with appropriate interpolating functions. Discuss your results: do they match your complexity analysis?

Yes, the plots confirm that the bubble sort algorithm has Quadratic time complexity which validates our derivation in question 1.