Algorithms – Q4 (20/11/23)

A list of n numbers is sorted by making passes through an algorithm.

To make a pass, compare the 1st and 2nd numbers. If necessary, swap them so that the 1st number is less than or equal to the 2nd number. Then repeat with the 2nd and 3rd numbers, and so on until the $(n-1)$st and nth numbers have been dealt with.

Repeat until a pass occurs with no swaps.

What are the minimum and maximum number of comparisons that are required?

Solution

If the numbers are already in increasing order, then $n - 1$ comparisons are required to complete the 1st pass, and the algorithms stops at this point, as there are no swaps.

So $n - 1$ is the minimum number of comparisons.

If the largest number is in the 1st position, and the smallest number is in the last position, then $n - 1$ passes will be required to move the smallest number to the 1st position, and each of these passes involve $n - 1$ comparisons. [Note that, in contrast to the Bubble Sort algorithm, the last number isn't removed after each pass.] Each of these passes will involve a swap, and so a further pass (involving $n - 1$ comparisons) will be required. This gives a total of $n(n - 1)$ comparisons.

So $n(n - 1)$ is the maximum number of comparisons.

Example 1: 5231 [Note that the numbers need not be in decreasing order]

[ ] indicates a comparison with a swap

( ) indicates a comparison without a swap

1st pass: [25]31, 2[35]1, 23[15]

2nd pass: (23)15, 2[13]5, 21(35)

3rd pass: [12]35, 1(23)5, 12(35)

4th pass: (12)35, 1(23)5, 12(35)

So $4 \times 3 = 12$ comparisons

Example 2: 5213 (where the smallest number is not in the last position)

1st pass: [25]13, 2[15]3, 21[35]

2nd pass: [12]35, 1(23)5, 12(35)

3rd pass: (12)35, 1(23)5, 12(35)