# Homework 3: Bubble sort and insertion sort

1. The numbers 15, 73, 29, 66, 35, 11, 43, 21 held in array **a** of length n are to be sorted using a bubble sort. The array index starts at 1.

The algorithm is given below.

n = length(a)

repeat

flag = False

for count = 1 to n-1

if a[count] > a[count + 1] then

temp = a[count]

a[count] = a[count + 1]

a[count + 1] = temp

flag = True

endif

next count

n = n-1

until n = 1 OR flag = False

(a) Show the state of the array after each pass through the data. [6]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **15** | **73** | **29** | **66** | **35** | **11** | **43** | **21** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **15** | **29** | **66** | **35** | **11** | **43** | **21** | **73** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **15** | **29** | **35** | **11** | **43** | **21** | **66** | **73** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **15** | **29** | **11** | **35** | **21** | **43** | **66** | **73** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **15** | **11** | **29** | **21** | **35** | **43** | **66** | **73** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **11** | **15** | **21** | **29** | **35** | **43** | **66** | **73** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **11** | **15** | **21** | **29** | **35** | **43** | **66** | **73** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

(b) What is the purpose of the variable **flag** in the algorithm? [2]

To check if an item has been swapped during the pass, this avoids unnecessary passes once the array is already sorted. Also tells us that the list is sorted

(c) How many times will the REPEAT loop be performed when sorting the following numbers? Explain your answer. [2]

32, 11, 15, 21, 29, 35, 43, 66, 73

32 is the only item in the wrong position, so it will be swapped with it’s next item 4 times, so the repeat loop runs twice because one pass to sort 32, and one pas thes to check that the data is fully sorted

2. (a) The pseudocode below sorts the array aList, of length n. Array indexing starts at 0.

for i = 1 to n – 1

currentvalue = aList[i]

pointer = i - 1

while pointer >= 0 and alist[pointer] > currentvalue

alist[pointer + 1] = alist[pointer]

pointer = pointer – 1

endwhile

alist[pointer + 1] = currentvalue

next i

Complete the empty (unshaded) cells in the trace table for an execution of the algorithm when the array aList contains the values 7, 3, 6, 2 in that order.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **aList** | | | |
| **n** | **i** | **currentvalue** | **pointer** | **[0]** | **[1]** | **[2]** | **[3]** |
| **7** | **3** | **6** | **2** |
| 4 | 1 | 3 | 0 |  | 7 |  |  |
|  | 2 |  | -1 | 3 |  |  |  |
|  | 2 | 6 | 1 |  |  | 7 |  |
|  | 3 |  | 0 |  | 6 |  |  |
|  | 3 | 2 | 2 |  |  |  | 7 |
|  |  |  | 1 |  |  | 62 |  |
|  |  |  | 0 |  | 3 |  |  |
|  |  |  | -1 |  |  |  |  |

[8]

(b) State the name of the above algorithm

Insertion sort [1]

(c) What is the order of time complexity of the algorithm?

[1]

O(n^2)

Total 20 marks