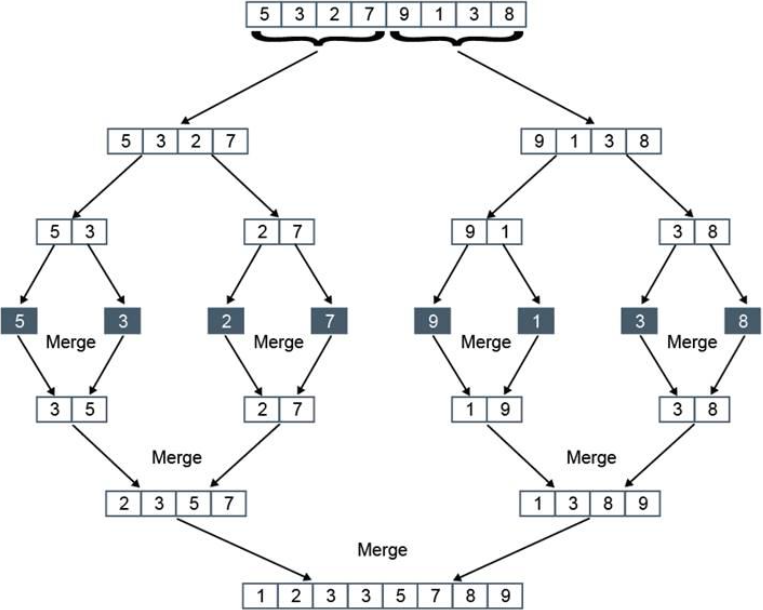
# Homework 3 Searching and sorting

1. (a) The diagram below shows an algorithm in graphical form. What is the algorithm depicted? [1]



(b) Describe briefly how the algorithm works. [5]

(c) Which of the following is the order of time complexity of the algorithm?

O(log2n) O(n2) O(n log2n) O(n) [1]

2. (a) Give **three** essential properties of a recursive routine. [3]

(b) Which of the following subroutines are recursive (i.e. fully implement recursion correctly)? [3]

(i) SUB addNumbers(num,x)

IF num > 1

x = num + addNumbers(num – 1, x)

ENDIF

RETURN x

ENDSUB

(ii) SUB printNumbers(num)

OUTPUT (num)

RETURN num + printNumbers(num – 1)

ENDIF

ENDSUB

(iii) SUB numbers(num)

IF num <= 0 THEN

RETURN 0

ELSE

RETURN numbers(num) - 1

ENDIF

ENDSUB

(c) The following pseudocode subroutine merges two sorted lists of equal or unequal lengths. It is called with the two lists shown in the main program.

NOTE: Given that list1 = [1, 2, 5, 10], then list1[1:] = [2, 5, 10].

That is, it returns a sublist starting with list1[1].

1. SUB merge (list1,list2)
2. IF length(list1) = 0 and length(list2) = 0
3. return []
4. ENDIF
5. IF length (list1) = 0
6. return list2
7. ENDIF
8. IF length (list2) = 0
9. return list1
10. ENDIF
11. IF list1[0] <= list2[0]
12. return [list1[0]] + merge(list1[1:], list2)
13. ENDIF
14. IF list2[0] <= list1[0]
15. return [list2[0]] + merge(list1, list2[1:])
16. ENDIF
17. ENDSUB
18. #main program
19. list1 = [3, 7, 9]
20. list2 = [1, 2, 5, 10]
21. mergedList = merge(list1,list2)
22. print(mergedList)

(i) What will be printed when the statement print(mergedList) at line 23 is executed? [1]

(ii) Give the statement numbers of the recursive calls in the subroutine. [2]

(iii) What parameters will be passed to the subroutine the first, second and third times that a recursive call is made? [3]

First time:

Second time:

Third time: