

Data Structures (IT205) 2014-15

Re-exam

19th November, 2014

Time: 2 hours

marks: 60

This question paper consists of 3 questions printed on a single page back-to-back. Check that your question paper is complete. Each question is worth 20 marks.

1. (a) Can the same data item move both left and right during the course of execution of merge-sort on an input consisting of n distinct numbers? Justify your answer.
(b) When executing merge-sort on an array $A[1..n]$ of distinct elements, element $A[i]$ moves left if and only if $A[i]$ is NOT the ----- among -----.
(c) Modify the code of bubble sort so it sorts n distinct elements into an alternating sequence. This means that the first element is less than the second and the second element is greater than the third and the third element is less than the fourth etc.
2. DS_1 : Suppose you are given a queue of five stacks each stack having the capacity of five integers. The insert operation in this two level data structure is such that an insert is like an enqueue (back most position of the queue), however each position being a stack of five integers, if that stack is full then the enqueue happens in the next stack. For the delete operation, the dequeue happens at the head of the queue and the corresponding stack's topmost element is returned.

DS_2 : One can similarly define a stack of five queues each queue having a capacity of five integers. Give a permutation of the elements $\{1, \dots, 25\}$ such that if you insert the elements in this order into DS_1 , then delete them and in insert them into DS_2 (in the order in which they are deleted from DS_1 and then delete them from DS_2 the order in which they are deleted is:

1, 25, 24, 2, 3, 23, 22, 4, 5, 21, 20, 6, 7, 19, 18, 8, 9, 17, 16, 10, 11, 15, 14, 12, 13

3. Procedure(L, key)

```
1.  $x \leftarrow HEAD[L]$ 
2. WHILE ( $x \neq null$ )
3.     IF ( $key > key[x]$ )
4.         THEN  $key \leftarrow key - 15$ 
5.          $x \leftarrow PREV[x]$ 
6.     ELSE  $key \leftarrow key + 15$ 
7.          $x \leftarrow NEXT[x]$ 
```

- (a) Does this code always, sometimes or never go into infinite loop.
- (b) If the answer is always or never, then justify. If the answer is sometimes then explain one with an input for which it goes into infinite loop and one for which it does not.

NOTE: The linked list L is assumed to have at least 5 elements. Do not give answers for degenerate/trivial cases like an empty list or a list with fewer than five elements. The input key to the procedure is an integer and the key values of the nodes in the linked list are also integers.