Data Structures (IT205)
Final Exam

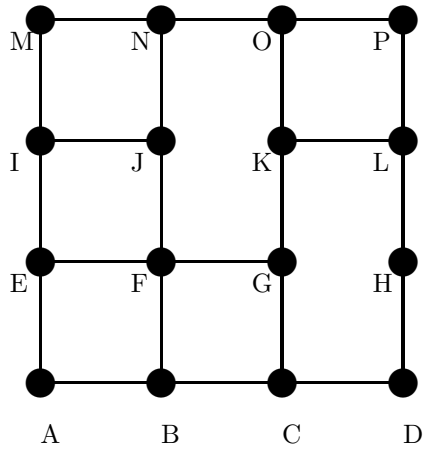
26^{th} November, 2012

Time: 2 hours **marks: 80**

This paper contains FIVE questions printed on 2 back-to-back pages. Check that you have it complete. Answer ANY FOUR of the five questions. Each Question is of 20 marks. NO QUERIES.

1. (a) For the graph drawn below, write down the adjacency list representation. Use the **lexicographic order** for the order of vertices in the vertex array and also for the linked lists representing the adjacencies.
   (b) Draw the BFS Tree for this graph assuming A as the source node. Repeat the procedure with K as the source node instead.
   (c) What is the maximum queue size in each case?
   (d) What is the maximum number of children of any node in the two cases?
   (e) Mention the cross-edges in each case. Are there any horizontal cross-edges?

   [**WARNING: If you use some other order (different from the lexicographical ordering A,B,C,...) your answers for the ENTIRE question will be marked WRONG and you will get ZERO**].



2. For the graph drawn above:
   (a) For the graph drawn below, write down the adjacency list representation. Use the **lexicographic order** for the order of vertices in the vertex array and also for the linked lists representing the adjacencies. (If you are attempting question 1 also, then no need to repeat this part).
   (b) Draw the DFS tree. What is the maximum number of children for any node in this DFS tree?
   (c) Which node achieves the highest difference between discovery time d and finish time f?
   (d) Which nodes have more than 1 child in the DFS tree?
   (e) In the DFS tree classify the non-tree edges as back edges and cross-edges.

   [**WARNING: If you use some other order (different from the lexicographical ordering A,B,C,...) your answers for the ENTIRE question will be marked WRONG and you will get ZERO**].

3. Suppose in the merge-sort procedure, instead of recursively calling merge-sort, you simply apply only the merge step repeatedly. Will the procedure sort any input sequence correctly? We may assume that one repeats merging in a while loop which terminates when during any iteration, the entire left subarray gets written before the entire right subarray. Is there any input array for which the algorithm will enter an infinite loop?
[You may assume that the input array consists of distinct elements and it will ease your analysis if you assume that the elements are the numbers $1, \ldots, n$**.]**

4. Sorting algorithms **sort** sequences. Suppose we run these routines but have an exit clause which terminates the procedure the moment the array becomes a minimum binary heap, even if not sorted. For the following sequence state the array configuration at which quick-sort, insertion-sort, merge-sort and bubble-sort terminate. Are they identical? Are any of them the final sorted order?

   $2, 4, 3, 7, 1, 5, 8, 6$

   For the algorithms listed above which terminate only after sorting is complete, give another input order of the same 8 elements such that termination happens before sorting is complete. However the input sequence itself should not be a minimum binary heap, so that the algorithms need to change the configuration at least once before termination.

5. The tree drawn below is a Binary Search Tree (the key values satisfying the BST property are **NOT SHOWN**. The letters are merely names of the nodes, and the numbers represent a cost field different from the field used to index the binary search tree.) Perform a sequence of rotations so that the final tree is also a BST (Rotations always preserve the BST property) and such that in the resultant tree no node is an ancestor of another node which has lower cost. Write the whole sequence of rotations as (left/right, node label). Draw the intermediate tree after each rotate operation.