01OGD Algorithms and Programming

26/06/2017 – Part I: Theory (12 points)

1. (1 point)

Given the following sequence of pairs, where the relation i-j means that node i is adjacent to node j:

0-5, 1-3, 0-2, 4-0, 7-6, 3-4, 9-8, 1-3, 6-7, 10-2

apply an on-line connectivity algorithm with quick find, showing at each step the contents of the array and the forest of trees at the final step. Node names are integers in the range from 0 to 10.

2. (1 point)

Sort in ascending order with merge sort the following array of integers:

31 19 2 14 43 3 89 23 29 17 51 10 18 16 8 100

Show relevant intermediate steps.

3. (2 points)

Suppose to have an initially empty priority queue implemented with a heap. Given the following sequence of integers and * character:

20 12 9 17 * * 85 * 71 29 31 * * 14 * 41 27 38

where each integer corresponds to an insertion into the priority queue and character * corresponds to an extraction of the maximum, show the priority queue after each operation and return the sequence of values extracted. At the end, change the priority of 31 into 11 and draw the corresponding priority queue.

4. (1.5 points)

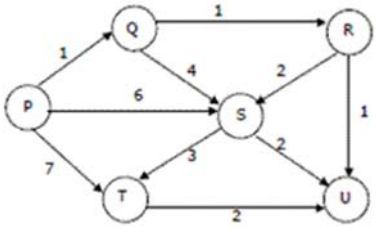
Write the following arithmetic expression in infix, prefix and postfix form visiting the corresponding binary tree: A * ((B / C) - (D * (E - F)))

5. (2 points)

Given the sequence of keys GFERBAOI, where each character is identified by its index in the English alphabet (A=1, ..., Z=26), draw the final configuration of an initially empty hash table of size 17 where insertion of the previous sequence character by character occurs. Assume open addressing with quadratic probing with $c_1 = 1$ and $c_2 = 1$. Show relevant intermediate steps.

6. (2.5 points)

Consider the DAG: starting from **P**, visit it in topological order and redraw it in topological order. Whenever necessary consider nodes in alphabetical order. Assume an alphabetical order for the adjacency list.



7. (2 points)

On the following directed and weighted graph, find all shortest paths connecting node **A** with all the other nodes resorting to Dijkstra's algorithm. If necessary, consider nodes and edges in alphabetical order.