01OGD Algorithms and Programming

26/01/2016 – Part I: Theory (12 points)

1. (2 points)

Given the following sequence of integers stored in an array:

4 53 92 52 32 34 13 12 91 93 22

perform the first 2 steps of quicksort to obtain an ascending order. At each step indicate the pivot element you selected. NB: steps must be improperly considered in breadth on the recursion tree, rather than in depth. Return as a result the 2 partitions of the original array and the two partitions of the two partitions found at the previous step.

2. (2 points)

A priority queue is implemented as a heap of data. Data are pairs of intergers, where the latter one represents the priority. The root of the heap stores the maximum priority. Insert the following sequence of data into the initially empty priority queue:

(1,20) (4,32) (5,19) (3,51) (7, 28) (8,74) (9,9) (0,81) (10,17) (6,41) (2,37)

Show the priority queue after each insertion. At the end, perform in sequence 2 extractions of the maximum. Show at each step the resulting priority queue.

3. (2 points)

Consider the sequence of keys "alpha", "beta", "delta", "epsilon", "zeta", "eta", "theta", "iota", "kappa". Suppose h("alpha")=2, h("beta")=14, h("delta")=18, h("epsilon")=17, h("zeta")=17, h("eta")=16, h("theta")=18, h("iota")=2, h("kappa")=14. Show an initially hash table of size 19 after the insertion of the sequence of keys. Use open addressing with linear probing.

4. (2 points)

Using a greedy algorithm find an optimal Huffman code for the following symbols with specified frequencies:

a: 46 c: 20 g: 14 k: 3 i: 35 n: 13 o: 24 r: 19 t: 12 u: 17

5. (2 points)

Insert at the root of an initially empty BST the following sequence of keys:

12 34 7 89 23 45 33 56

6. (2 points)

Consider the DAG: starting from **a**, 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.