Aalto University School of Science and Technology — EXAMINATION
Department of Computer Science — 7.12.2015
CSE-A1140 Data Structures and Algorithms

Write on each paper your name, student number, degree programme, and the course code with name. Also write the date, hall, the number of papers you return, and your *signature*. Using any extra devices is prohibited in this examination.

1) Ten Questions (10 x 1p + 1p)

This is a compulsory part of the final exam. You need to get at least 5p out of the maximum 10p so that the rest of the exam will be checked. However, this part alone is not enough to pass the whole exam. On the other hand, in order to get 5p, you are not required to give "the exactly correct answer", but more or less show that you have understood the functionality of the code fragments related to this part. Thus, pay attention to the reasoning. Refer to the code line numbers if possible.

In the following, you can see two algorithms (`bs1` and `bs2`) searching for an item `x` from a sorted array (`table`). Read through all the questions below without answering them and after that familiarize yourself with the code throughout. After this, answer all the questions and take time for pondering and explaining your reasoning. Note, however, that all the questions refer to the given algorithms. In addition, the claims in the questions can be justified to be either true or false, thus the *argumentation* is the only thing that matters for the points.

```
1  int bs1(int table[], int x) {
2      int low = 0;
3      int high = table.length - 1;
4      int mid;
5
6      while( low <= high )
7      {
8          mid = (low + high) / 2;
9
10         if (table[mid] < x)
11                 low = mid + 1;
12         else if (table[mid] > x)
13                 high = mid - 1;
14         else return mid;
15     }
16     return -1;
17 }
```

```
18 def bs2(table, x, low, high):
19     if low > high:
20         return -1
21     mid = (low + high) / 2
22     item = table[mid]
23     if item == x:
24         return mid
25     elif x < item:
26         return bs2(table,x,low,mid-1)
27     else:
28         return bs2(table,x,mid+1,high)
29
30
31
32
33
34
```

a) *Describe* how Algorithm `bs1` works (without an example). Note! Try to answer how it solves the computational problem – do not just explain the code line-by-line.
b) *Describe* how Algorithm `bs2` works. How this differs from the previous one?
c) *Give an example of Algorithm 1* in case we are searching for the item `x` = 512 from an array comprising the items 1, 2, 4, 8, 16, 32, 64, 128, 256, and 512. Hint! Show in tabular form the changes in variables `low`, `high`, and `mid` during the execution of the algorithm. What is the return value of the execution, and what is the result of the computation in this case?
d) *Give an example of Algorithm 2* in case we are searching for the element `x` = 100 from the aforementioned array. Hint! Show in tabular form the changes in variables `low`, `high`, `mid`, and `item` as before. What's the output in this case?
e) Determine the *input size n* of Algorithm 1 and 2, *i.e.*, which variables and how the time complexities of the algorithms depend on?
f) *Analyze* the time complexity of Algorithm 1 in terms of the input size *n*.
g) *Analyze* the time complexity of Algorithm 2 in terms of the input size *n*.
h) Algorithm 1 was tested with a large data set (search for the smallest element). The running time was estimated to be 1 millisecond. After this, the data set was duplicated, and the running time was measured to be 2 milliseconds. Estimate the running time if the data set would be duplicated again. Justify your answer.

i) What kind of assumptions and boundary conditions the correct and efficient execution of Algorithm 2 sets for the array `table`, and the items it contains?
j) *Argue* whether it is true or false: Algorithm 1 is more efficient than Algorithm 2.
Bonus question:
k) *Ponder and compare* the memory consumption of Algorithm 1 and 2.

2) Terminology (2p + 2p + 2p + 2p)

Define the following *concepts* (4 x 1p). In addition, *give an example* of each (4 x 1p).

a) Heap property (kekoehto)
b) Hash function (hajautusfunktio)
c) Linear probing (lineaarinen kokeilu)
d) Selection problem (valikointi-ongelma)

3) Tree traversal (2p + 2p)

A binary tree can be uniquely determined by its preorder (*esijärjestys*) and inorder (*sisäjärjestys*). Consider a tree having the preorder K-I-B-A-M-H-L-P-Q-F, and the inorder B-I-M-A-H-K-P-L-Q-F. *Draw* the tree, and *give* the corresponding *postorder (jälkijärjestys)*.

4) Sorting (3p + 3p + 2p)

You need to choose an algorithm for a task in which a certain data must be put in sorted order. What things (criteria) influence your decision? Read first the whole assignment.

a) *Choose three* (3) essential *criteria* in light of which you examine the task. *Argue* why and how your criteria are related to sorting problem.
b) *Name* at least one *algorithm* for each criterion that satisfies and do not satisfy the criterion (there is no need to explain the operational principles of the algorithms). Give this answer in a matrix in which columns (3) have criteria and below are rows (2) that have the names of the algorithms that satisfy and do not satisfy each criterion.
c) *Name an algorithm* that satisfies **all** the criteria chosen by you. *Name* also *an algorithm* that do not satisfy two of the criteria.

5) Graphs (4p + 4p + 4p)

Define the following concepts. *Give a* mutually distinct *example* of each. In your example, use a connected graph (*yhtenäinen verkko*) that has at least 7 nodes and at least 10 edges. In addition, *describe* briefly how the corresponding computational problem can be solved (use your examples, describe an initial condition, mention an algorithm that can take it as an input, and describe briefly how this algorithm solves the problem).

a) Spanning tree (virityspuu)
b) Minimun spanning tree (minimaalinen virityspuu)
c) Shortest paths spanning tree (lyhimmin poluin virittävä puu)