Aalto University, School of Science
Department of Computer Science
Tommi Junttila (050-4300861)

CS-A1140 Data structures and algorithms — Autumn 2016

Exam December 12th, 2016, at 13:00–16:00

Use of calculators is not allowed in the exam.
Note: your answers should be informative, well structured and concise (no long essays).

1. (a) A fixed-size mutable array (Array in Scala) does not support efficient insertion of new elements at the end of the array. But mutable *resizable arrays* (ArrayBuffer in Scala) support such operations in amortized constant time — explain how this is done and what "amortized constant time" means (proofs with recurrences are not required).

   (b) A *mutable queue* data structure supports the following operations:

   - enqueue(e), which adds the element e at the end of the queue, and
   - dequeue(), which removes and returns the first element in the queue.

   Describe how such a data structure can be implemented so that both of these operations can be performed in constant time.

   10 points

2. Define the following concepts:

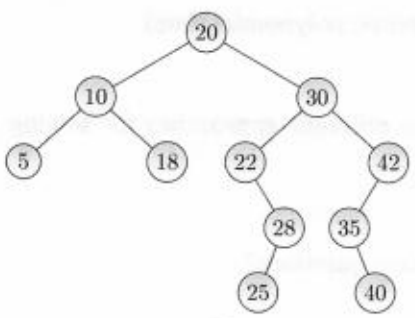
   - a sorting algorithm that works in-place
   - a stable sorting algorithm

   Describe *one* of the following algorithms briefly either in pseudo-code or verbally in a clear and unambiguous way: (a) insertion sort, (b) merge sort, or (c) quicksort.

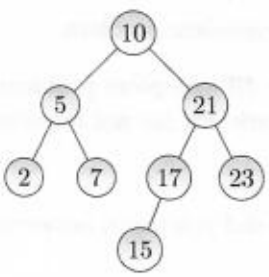
   For the chosen algorithm, also answer the following questions (justify your answer with at most few sentences): (i) Is the algorithm stable? (ii) Does the algorithm work in-place? (iii) What is the best-case running time of the algorithm? (iv) What is the worst-case running time of the algorithm? In the last two questions, denote the length of the array to be sorted with n.

   10 points

3. Consider the binary search tree A shown below (the keys are integers and they are drawn inside the nodes). Perform the following operations (in the given order!) to the binary search tree and, after each operation, illustrate the resulting binary search tree: (i) insert the key 17, (ii) remove the key 18, and (iii) remove the key 30.



tree A



tree B

Describe an algorithm for listing all the keys in a binary search tree in ascending order. Assume that a binary search tree has n nodes and height h. What is the worst-case running time of the algorithm with respect to these parameters?

Define what is required of a binary search tree so that it is an AVL tree. Is the binary search tree A above an AVL tree? Justify your answer.

Consider the AVL tree B above. Insert the key 12 to it in a way that preserves the AVL property; describe what kind of transformation your insertion applies and why.

10 points

4. A subsequence of a string is a sequence of characters that occur in the same order in the string (i.e., a sequence obtained by removing some of the characters from the string). Now consider the following computational problem:

   - Given a string s of n characters.
   - What is the length of the longest subsequence of s that is a palindrome?

   For instance, if s is "character", then the answer is 5 as "carac" is a palindrome subsequence of "character". On the other hand, if s is "acabbdea", then the answer is 4 because of "abba".

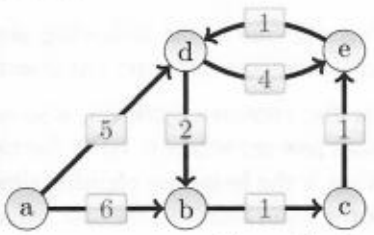
   Describe a dynamic programming algorithm for solving the problem. It should work in time $O(n^2)$ and use $O(n^2)$ amount of extra memory. Apply your algorithm to the string "dcaadc".

   10 points

5. Explain (with pseudo-code or with a very clear and structured verbal description) Dijkstra's algorithm for finding shortest paths in edge-weighted directed graphs with non-negative edge-weights. What kind of data structures are needed in the algorithm?

   Describe/illustrate how the algorithm works on the graph below when the source vertex is the vertex a.

   Given an edge-weighted directed graph $G = (V, E, w)$ with $|V| = n$ and $|E| = m$, what is the worst-case running time of the algorithm? Justify your answer with few sentences.



   Explain (perhaps with a simple example) why the algorithm does not necessarily work when the edge-weights can be negative.

   10 points

6. Define the following concepts:

   - A polynomial-time solvable decision problem.
   - A decision problem in **NP** (non-deterministic polynomial time).
   - An **NP**-complete problem.

   Describe one **NP**-complete problem and two different approaches for solving it in practise (detailed algorithms are not required).

   9 points

7. At what time did you finish answering the exam questions?

   1 points