TDT4120 Algorithms and Data Structures

Examination, November 24, 2020, 09:00–13:00

Academic contact Magnus Lie Hetland
Support material code A

Problems

1 Explain Huffman's algorithm (HUFFMAN) in your own words. It is sufficient to describe which problem it solves and which steps it performs; you need not explain why it produces a correct answer or discuss running time. Which problem does the algorithm solve?

Explain and elaborate. Link to relevant theory, possibly in different parts of the curriculum.

2 Several algorithms to find minimum spanning trees are based on the same principle for selecting one edge at a time. Explain what they require of the edges that are selected and give a high-level explanation for why this is safe.

Explain and elaborate. Link to relevant theory, possibly in different parts of the curriculum.

3 Discuss different running time functions (e.g., in the best and worst case, and various kinds of averages) and the relationship between these and different kinds of asymptotic notation.

Explain and elaborate. Link to relevant theory, possibly in different parts of the curriculum.

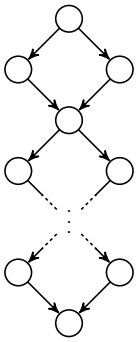
4 Your friend Smartnes has come across the algorithm SELECTION-SORT, which she describes at follows:

> Your input is an array $A[1..n]$. For $i = n$ down to 1, switch the positions of $A[i]$ and the largest element of $A[1..i]$.

She intends to improve the algorithm by using a binary search tree to find the largest elements. Discuss the advantages and disadvantages of this approach, and what it might be natural to do instead.

Explain and elaborate. Link to relevant theory, possibly in different parts of the curriculum.

5 The figure to the right shows the decomposition of an instance into subinstances (also known as subproblems). Each edge points from an instance to a subinstance.



Discuss the running time of a naïve recursive solution to the problem, and how it may be improved. What is this improvement called and what is it about the problem structure that makes the improvement possible?

Explain and elaborate. Link to relevant theory, possibly in different parts of the curriculum.

6 Discuss the difference between QUICKSORT and RANDOMIZED-QUICKSORT, and how this difference differs from the difference between SELECT and RANDOMIZED-SELECT. How could the situation change if you modify QUICKSORT to use SELECT as a subroutine?

Explain and elaborate. Link to relevant theory, possibly in different parts of the curriculum.

7 You have a database of rules of the type "If A then B" for various logical variables A, B, C, etc. The system supports so-called forward chaining: If it is told that a variable X is true, then the consequences of X are set to be true, and then the consequences of these, etc., until all consequences are found.

Your boss would like a new functionality, where you get to know *why* a variable is true: Which other variable was it a consequence of? And which variable was *that* a consequence of, etc.? She also wants these explanations to be as simple as possible, i.e., avoid chains with an excessive number of variables. How would you approach this problem?

Explain and elaborate. Link to relevant theory, possibly in different parts of the curriculum.

8 Your friend Lurvik has read that FORD-FULKERSON has a pseudopolynomial running time, and believes this is because the maximum flow problem is actually NP-hard. He argues as follows: Maximum flow is a special case of so-called min-cost hyperflow, where the edges have weights and each edge may be connected to multiple nodes—and in the article *A Hypergraph Network Simplex Algorithm*, Beckenbach (2018) writes:

> In particular, it is NP-hard to find an integral min-cost hyperflow (e.g. by reduction to 3D-Matching) [. . .].

Discuss Lurvik's claims and Beckenbach's statements.

Explain and elaborate. Link to relevant theory, possibly in different parts of the curriculum.