TDT4120 Algorithms and Data Structures

Examination, November 28, 2019, 09:00–13:00

Academic contact	Magnus Lie Hetland
Support material code	D

Problems

5 % **1** Consider the following statement about DIJKSTRA:

> It permits negative edge weights, unlike BELLMAN-FORD.

Is this correct? Answer yes or no and explain briefly.

5 % **2** Consider the following statement about INSERTION-SORT:

> It has a best-case running time of $\Omega(n \lg n)$.

Is this correct? Answer yes or no and explain briefly.

5 % **3** Consider the following statement about divide-and-conquer:

> The method should be avoided with overlapping subproblems.

Is this correct? Answer yes or no and explain briefly.

5 % **4** Consider the following statement about TOPOLOGICAL-SORT:

> The vertices are sorted by decreasing discover-time.

Is this correct? Answer yes or no and explain briefly.

5 % **5** Consider the following statement about heaps:

> BUILD-MAX-HEAP has a running time of $\Theta(n \lg n)$.

Is this correct? Answer yes or no and explain briefly.

5 % **6** Which algorithms in the curriculum solve the single-source shortest path problem in weighted, directed graphs?

Do not include algorithms that solve the all-pairs shortest path problem.

5 % **7** Which algorithms in the curriculum find minimum spanning trees? (We assume weighted, undirected, connected graphs, here.)

5 % **8** The following partially redacted lemma is taken from a section in the textbook that shows that an algorithm in the curriculum is correct.

> **Lemma** [redacted]
>
> Let C be an [redacted] in which each [redacted] $c \in C$ has frequency *c.freq*. Let x and y be two [redacted] in C having the lowest frequencies. Then there exists an optimal [redacted] for C in which the [redacted] for x and y have the same length and differ only in the last [redacted].

Which algorithm is discussed, and which problem does it solve?

5 % **9** Your friend Smartnes has given you the following array, which she claims is a max-heap:

$A = \langle 7, 3, 8, 6, 5, 9, 4, 2 \rangle$

You do not agree that this is a heap, but are still willing to perform Heap-Extract-Max on the array, *even though the result will be incorrect*. (In other words, you are to execute the steps of Heap-Extract-Max, without correcting A in any way.)

What does A look like afterward?

(Give only the first 7 elements of A after the operation.)

5 % **10** What is $O(n) + \Omega(n) + \Theta(n) + o(n) + \omega(n)$?

5 % **11** Solve the following recurrence exactly, where n is a positive integer:

$$\begin{aligned} T(1) &= 1 \\ T(n) &= T(n-1) + 2^{n-1} && \text{(if } n > 1\text{)} \end{aligned}$$

Give your answer *without* using asymptotic notation.

5 % **12** Your friend Klokland has designed a version of Merge with a running time of $\Theta(n^2)$. If you use Klokland's version instead of the ordinary one, what is the running time of Merge-Sort? Give your answer in Θ notation. (A brief explanation is optional.)

5 % **13** In a flow network, if you have residual capacities $c_f(u,v) = 9$ and $c_f(v,u) = 3$, and $f(u,v) > 0$, what are $f(u,v)$ and $c(u,v)$? Give the values in the ordinary way, as two numbers with a slash between them, like 1/2 if you mean $f(u,v) = 1$ and $c(u,v) = 2$.

5 % **14** Consider the following statement about FLOYD-WARSHALL:

> If $d_{ij}^{(k)}$ is set to $d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$ then $\pi_{ij}^{(k)}$ is set to $\pi_{ik}^{(k-1)}$.

Is this correct? Answer yes or no and explain briefly.

5 % **15** Consider the following statement about minimum spanning trees in a weighted, undirected graph $G = (V, E)$ with all distinct edge weights:

> If the vertices V may be partitioned into the sets X and Y (without intersection) then the minimum spanning tree will necessarily contain the edge between X and Y with the lowest weight.

Is this correct? Answer yes or no and explain briefly.

(It may not be enough to describe what some curriculum algorithm does.)

5 % **16** Your friend Lurvik claims he has designed a dynamic programming algorithm that puts together the coolest outfit possible from the clothes in his wardrobe. When you inspect the algorithm, you notice that it merely computes the coolness degree of the coolest outfit, *without actually determining which articles of clothings are involved!* He thinks this is a trivial distinction, and that adding the functionality would be easy. What do you think? Explain briefly.

5 % **17** Your friend Gløgsund wishes to determine whether a graph has a minimum spanning tree of weight at most k, and thinks she has been able to reduce the problem in polynomial time to the traveling-salesman problem. Your other friend Klokland can't quite follow her proof, but believes it cannot possibly be correct. Do you think it's possible? Explain briefly.

5 % **18** Assume that you have a directed graph $G = (V, E)$ with positive integer edge weights, where $s, t \in V$. This graph may have multiple simple paths from s to t with a mimimum length, that is, multiple "shortest" paths. How would you find the one among these that consists of the *most* edges?

5 % **19** You are given three sequences $A = \langle a_1, \dots, a_m \rangle$, $B = \langle b_1, \dots, b_n \rangle$ and $X = \langle x_1, \dots, x_{m+n} \rangle$ and wish to decide whether X consists of A and B interleaved, i.e., that X consists of the elements of A og B, in their original order, but interleaved in some way, so, e.g.,

$$X = \langle a_1, a_2, b_1, a_3, b_2, b_3, a_4, \dots, b_n, a_{m-1}, a_m \rangle .$$

You may think of such an interleaving as first concatenating A and B to $\langle a_1, \dots, a_m, b_1, \dots, b_n \rangle$ and then (possibly) modifying the order of the elements (without adding or removing elements, and where a_i still comes before a_{i+1} and b_i still comes before b_{i+1}). It is quite possible that A and B contain some identical values, and that these occur multiple times.

Describe an algorithm that solves the problem.

5 % **20** You have a game board, represented as a directed graph $G = (V, E)$, with a set of k starting positions $A \subset V$ and a set of k ending positions $Z \subset V$ given. You also have k gaming pieces that are to begin in the starting positions A (one piece in each position). You are to move the pieces along the edges in G such that eventually there is one piece in each ending position Z. (It does not matter which piece ends up in which ending position.)

You are to execute this movement as a series of n moves. In each move you may move as many pieces as you wish (simultaneously), but they may at most move one step each (i.e., along a single edge), and you may not move two pieces to the same position (vertex) at the same time. You may assume that $k \geqslant 1$ and that it is possible to find a solution.

Describe an algorithm that solves the problem.