NTNU

Department of Computer and Information Science

Examination paper for TDT4120 Algorithms and Data Structures

Academic contact during examination	Magnus Lie Hetland
Phone	918 51 949
Examination date	December 4, 2017
Examination time (from–to)	09:00–13:00
Support material code	D
Other information	The problem sheets are handed in, with answers in answer boxes under the problems
Language	English
Number of pages (front page excluded)	6
Number of pages enclosed	0

Informasjon om trykking av eksamensoppgave

Originalen er

1-sidig ☑ **2-sidig** ☐

sort/hvit ☑ **i farger** ☐

Skal ha flervalgskjema ☐

Quality assured by Pål Sætrom

Checked by

Date Signature

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer these.

Please read this thoroughly

(*i*) Read the entire exam thoroughly *before* you start!

(*ii*) The teacher will normally take one round through the exam hall. Have your questions ready!

(*iii*) Write your answers in the answer boxes and hand in the problem sheet. Feel free to use a pencil! Or draft your answers on a separate piece of paper, to avoid strikouts, and to get a copy for yourself.

(*iv*) You are permitted to hand in extra sheets if need be, but the intention is that the answers should fit in the boxes on the problem sheets. Long answers do not count positively.

Problems

(5%) 1. (a) Which traversal algorithm in the curriculum finds shortest paths in unweighted graphs?

(5%) (b) One key ingredient in dynamic programming is optimal substructure. What is the other?

(5%) (c) What is the worst-case running time of RANDOMIZED-SELECT?

(5%) (d) RADIX-SORT sorts by one digit at a time. In which order?

(5%) (e) What does the graph have if BELLMAN-FORD fails (i.e., returns FALSE)?

(5%) (f) What does the attribute $v.\pi$ represent in (e.g.) PRIM and DIJKSTRA?

(5%) (g) What's the circuit input in the NPC proof for CIRCUIT-SAT (i.e., y, where $\mathrm{C}(y) = \mathrm{A}(x, y)$)?

2. In the following problems, make sure not to "throw away" information during simplification. Give tight bounds and use the most precise asymptotic notation you can (from among O, Ω and Θ).

(5%) (a) Simplify the expression $\Theta(n) + O(n^2) + \Omega(n^3)$.

(5%) (b) Simplify the expression $\Theta(n + \sqrt{n}) + O(n^2 + n) + \Omega(n \cdot (n + \lg n))$.

(5%) (c) Solve the recurrence $T(n) = T(n/3) + \lg n$. Give your answer in Θ notation.

3. Your friends Smartnes and Lurvik have designed one algorithm each, both of which take a binary max-heap as input. The heap is represented as an array $A[1..n]$ in the usual way. All the elements are different and the heap is *complete*.[1]

(5%) (a) Lurvik's algorithm is based on the assumption that the $\lceil n/2 \rceil$ smallest elements always will be last (i.e., in the leaf nodes). Is that correct? Explain your answer briefly.

(5%) (b) Smartnes claims her algorithm uses the structure of the heap to build a balanced binary search tree for the elements of A in linear time. Do you think she could be right? Explain briefly.

[1] All internal nodes have two children and all leaf nodes are on the same level, i.e., $n = 2^k - 1$ for some integer $k \geq 1$.

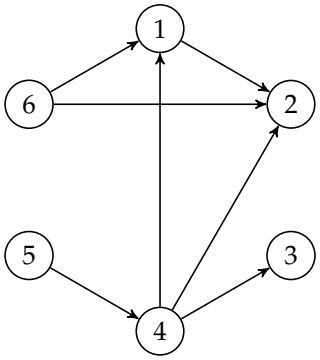


Figure 1: $T^{(3)}$ – used in problem 4

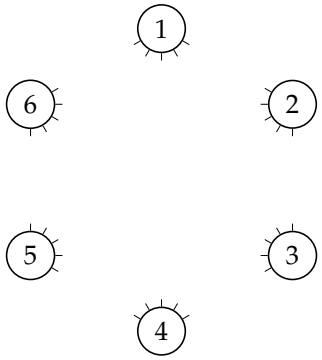


Figure 2: $T^{(4)}$ – answer to problem 4

(5%) 4. You are to simulate one iteration of TRANSITIVE-CLOSURE. In fig. 1 you see the directed graph corresponding to $T^{(3)}$. You are to compute $T^{(4)}$ and draw the corredponding directed graph.

Fill in the edges that are missing in fig. 2. Clearly indicate their direction.

(Note: Each $T^{(k)}$ is a completely new array. That is, we do not reuse the same one in each iteration.)

(5%) 5. (a) Draw a flow network with odd integer capacities but with an even maximum flow value.

(5%) (b) In a flow network where the capacities are even numbers, the maximum flow is even. Why?

(5%) (c) A team of researchers are to implement a set of projects.

- Each project consists of several tasks, and a certain number of the project's tasks (e.g., 7 of 12) must be performed. Different projects may have different numbers.
- Each task is to be performed by one researcher.
- Each researcher has a certain capacity, i.e., a number of tasks she has time to do.
- Each researcher is competent to do some of the tasks but not necessarily all.

You are to decide who performs which tasks, or determine that it can't be done.
Describe how you can solve this as a max-flow problem. Feel free to draw a flow network.

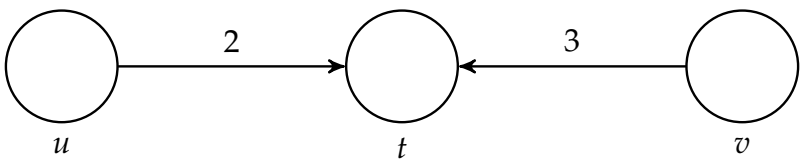
---

6. Your friend Klokland wants to solve the single-*destination* shortest-path problem in a directed graph $G = (V, E)$ with weight function $w : E \to \mathbb{R}$. He has changed RELAX so that it updates u based on $w(u, v)$ and $v.d$, rather than the other way around.

   He uses this new RELAX′ in a modified version of DFS: First he initializes the graph as usual, and when the recursive traversal returns to u after having visited v (and colored it black) he uses his RELAX′ on the edge (u, v). (If you wish, see pseudocode on p. 6 for elaboration.)

   Klokland claims that the algorithm works for directed, acyclic graphs. You are skeptical, and want to create a counter-example. The result is the graph shown below.

(3%) (a) What is the result of the algorithm if u is visited first? Fill in $u.d$, $t.d$ and $v.d$, below.



(7%) (b) Briefly explain how the algorithm can be modified so it's correct for (weighted, directed) acyclic graphs. (A very brief explanation will suffice.)

---

7. Given a game board as in fig. 3 (on p. 6), with $n \geq 3$ squares, you start at the far left and move toward the right. Each square indicates how far you can move in one jump when standing on that square. The goal is to reach the last square in as few jumps as possible.

   (You can assume that the input is such that it is always possible to reach the last square.)

(7%) (a) Describe an algorithm that efficiently determines how many jumps you must use. (You need *not* determine *which* jumps.) Feel free to describe it in your own words, or use formulas, figures, code or pseudocode.

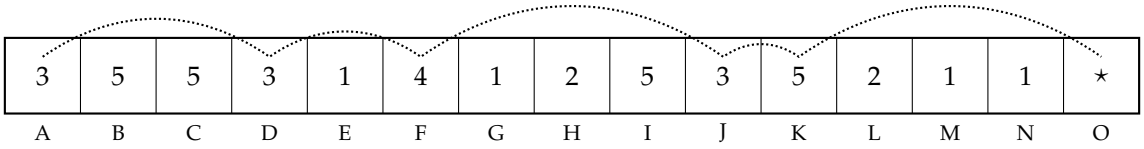


Figure 3: Example for problem 7. In the first jump, you can move from A to B, C or D. If you move to D you can then move to E, F or G in one jump. (Note: The example solution is not optimal.)

(3%) (b) What is the worst-case running time? Give your answer in Θ notation. Explain briefly.

Pseudocode for problem 6

The following is a more detailed description of the algorithm discussed in problem 6. INITIALIZE-SINGLE-SOURCE is as described in the textbook. The pseudocode is meant as a supplement, and it is possible to solve the problem without it. Note that you are *not* required to write pseudocode, or to refer to this code, in your answer.

```
RELAX'(u, v, w)
1  if u.d > v.d + w(u, v)
2      u.d = v.d + w(u, v)
3      u.π = v

DFS'(G, t)
1  INITIALIZE-SINGLE-SOURCE(G, t)
2  for each vertex u ∈ G.V
3      u.color = WHITE
4  for each vertex u ∈ G.V
5      if u.color == WHITE
6          DFS-VISIT'(G, u)

DFS-VISIT'(G, u)
1  u.color = GRAY
2  for each v ∈ G.Adj[u]
3      if v.color == WHITE
4          DFS-VISIT'(G, v)
5          RELAX'(u, v, w)
6  u.color = BLACK
```