Rules

Things you CAN use during the exam.
Any written or printed material is fine. Textbook, other books, the printed slides, handwritten notes, whatever you like. You may use electronic versions of the material, e.g. PDFs and the like.

In any case, it would be good to have a source for the relevant definitions, and also for notation, just in case you don't remember the precise definition of everything we discussed in the course.

Things you CANNOT use during the exam.
Any communication facility, other than to interact with the examiner.

The exam is ongoing from the moment you receive this document to the moment you submit your answers. **During that time, you must not communicate with anybody other than the examiner and you must not solicit or accept help from any third party with any part of the exam.**

Instructions

- This is a take-home exam. This document was sent to you electronically as a PDF.
- You can print out this document and write your answers in the appropriate spaces. If you do...
  - Fill out the first page and sign.
  - Put your personnummer and name on the subsequent pages in the header.
- If printing is not an option, you can answer the questions on empty sheets of paper. If you do...
  - Make sure to include, on your first page, your name, personnummer, and signature.
  - Mark every subsequent page with your personnummer and name.
  - You need not reproduce the left-most column of the tables in your answers. In that case, your answers will be matched to the questions in the table row by row, so make sure you match the order of your answers to that of the questions.
- Once finished, scan or otherwise photographically capture the pages and produce a PDF from them (using software such as Office Lens, for example).
- The **name of the PDF file** must be your personnummer followed by "`.pdf`", i.e. it has the format

  `yymmdd-nnnn``.pdf`
- **Return the PDF with your answers by replying to the email that you received the question sheet in**. The subject line must include "`[EDAA40 Exam]`" (without the quotes). Do not forget to attach the file.
- If you have questions for the examiner during the exam, contact him by phone first (you will receive the phone number in the email with the exam).

<u>Good luck!</u>

Programming contest

Were you member of a group that qualified for the EDAA40 programming contest this year?

	I was in a group that qualified but did not win.
	I was in the group that qualified and won.
	None of the above.

(please tick the appropriate box)

Group name (if applicable): ___________________________________

1 [15 p]

Suppose we have a tree (T, R) with root $a \in T$.

1. [5 p] Define a function $N : \mathbb{N} \longrightarrow \mathcal{P}(T)$ which computes, for every natural number n, the set of all nodes in T that have exactly n children.

   $N : \mathbb{N} \longrightarrow \mathcal{P}(T)$

   $n \mapsto \{t \in T : \#R(t) = n\}$

2. [5 p] Give an expression evaluating to $C_{\max}$, the maximal number of children of any node in the tree.

   $C_{\max} = \max\{\#R(n) : n \in T\}$

3. [5 p] Give an expression computing the node height H of the tree.

   H = $\max\{\# \ R^{-1}[\{n\}] : n \in T\}$

   Many solutions used a helper function to recursively compute the height, which is of course a valid answer (if done correctly).
   The solution above exploits the fact that the node height of a tree is the same as the size of the largest set of ancestors of a node (including the node itself!). One can also get there with the transitive closure of R^{-1}, but in that case special care must be taken to make sure that the result is the node height (by adding 1, for example).

2 [40 p]

Suppose we have a graph (V, E).
You can reuse previous answers. If they are incorrect, it will not affect the answer using them, i.e. I will assume you are using a correct answer to the previous question.

1. [5 p] A vertex $w \in V$ is said to be *reachable* from another vertex $v \in V$ iff there exists a path from v to w. (This includes empty paths, so every vertex is always reachable from itself.)
   Define the relation $R \subseteq V \times V$, such that $(v, w) \in R$ iff w is reachable from v.

   $R = E^+ \cup \{(v, v) : v \in V\}$

   The most common mistake on this task was omitting the second part, i.e. making sure that the reachability relation is reflexive.

2. [5 p] What properties does R have?
   Tick the corresponding box under "always", if R has the property for every graph (i.e., for every possible V and E defining a graph), under "never" if it does NOT have that property for any graph, and under "sometimes" if it has that property for at least one graph and does not have the property for at least one graph.
   Mark the corresponding box under "no answer" if you prefer to not give an answer. See below about how these tables are scored. (1 point per answer)

	always	**sometimes**	**never**	**no answer**
reflexive over V	X			
transitive	X			
symmetric		X		
antisymmetric		X		
asymmetric		X		

<u>About scoring these tables:</u>

Every correct answer **adds** the indicated number of points per answer, while every false answer **deducts** the same number of points. Marking "no answer" (or simply not marking any box in a given row) does not change the point score, so it counts as 0. Should the total score for the table be negative, it is counted as zero (0).

3. [5 p] Define the relation $M \subseteq V \times V$, such that $(v, w) \in M$ iff v is reachable from w and w is reachable from v:

   $M = R \cap R^{-1}$

   Many answers look more like $\{(v, w) \in V^2 : vRw \wedge wRv\}$, which is correct and a perfectly good answer.

4. [5 p] What properties does M have?
   Tick the corresponding box under "always", if M has the property for every graph (i.e., for every possible V and E defining a graph), under "never" if it does NOT have that property for any graph, and under "sometimes" if it has that property for at least one graph and does not have the property for at least one graph.
   Mark the corresponding box under "no answer" if you prefer to not give an answer. (1 point per answer)

   The correct answers to this table include M being reflexive, transitive, and symmetric, which means it is an equivalence relation. Realizing this can be helpful in answering the next question.

	always	**sometimes**	**never**	**no answer**
reflexive over V	X			
transitive	X			
symmetric	X			
antisymmetric		X		
asymmetric		X		

5. [10 p] A set of vertices $C \subseteq V$ is called a *strongly-connected component* of this graph iff (a) for any two vertices $v, w \in C$, v is reachable from w and w is reachable from v, and (b) it is maximal, i.e. there is no set C' with $C \subset C' \subseteq V$ such that for any two elements $v, w \in C'$, v is reachable from w and w is reachable from v.
   Define the set $S \subseteq \mathcal{P}(V)$ of **all** strongly-connected components of this graph.

   $S = V/M$

   As can be seen in the previous task, M is an equivalence relation, and the strongly connected components are its equivalence classes, so one way of defining S is as the quotient set of V and M.

   A common way of answering this was $\{M(v) : v \in V\}$, which is also correct.

6. [10 p] Suppose S is the set of strongly-connected components of some graph (V, E). Judge the truth of the following formulae.
   Tick the corresponding box under "always", if the formula is true for all graphs, the one under "never" if there is no graph for which this formula is true, and the one under "sometimes" if there is at least one graph for which it is true and at least one for for it is false.
   Mark "no answer" if you prefer not to give an answer. (2 points per answer)

	always	**sometimes**	**never**	**no answer**
$V = \bigcup_{a \in S} a$	X			
$\forall r \in S, s \in S \ (r \cap s = \emptyset)$		X		
$\forall r \in S, s \in S \ (\exists n \in r \cap s \ (n = n))$		X		
$\exists r \in S, s \in S \ (r \cap s \neq \emptyset)$		X		
$\forall s \in S \ (s \neq \emptyset)$	X			

3 [20 p]

Suppose we have an surjection $h : A \longrightarrow\!\!\!\!\rightarrow C$ which we know to be the result of composing $f : A \longrightarrow B$ and $g : B \longrightarrow C$, i.e. $h = g \circ f$, but we don't know those two functions, nor do we know B. Knowing that h is surjective, what can we say about f and g?

In the following, you will be given a few statements. You are asked to **decide whether they follow from the fact that h is surjective**. If you think a statement does follow from that, prove it. Otherwise, show a counterexample. A counterexample consists of definitions of A, B, C, and f, g, and h, such that the h is surjective, $h = g \circ f$, and the statement is not true. Use proper definitions, not pictures.

1. [5 p] f must be surjective (circle correct answer). YES NO
   If yes, prove it, if no, find a counterexample.

   $A = \{a\}$
   $B = \{b_1, b_2\}$
   $C = \{c\}$
   $f = \{(a, b_1)\}$
   $g = \{(b_1, c), (b_2, c)\}$
   $h = \{(a, c)\}$

2. [5 p] g must be surjective (circle correct answer). YES NO
   If yes, prove it, if no, find a counterexample.

   To show: $g \circ f$ surjective implies that g is surjective, i.e.
   $h(A) = g \circ f(A) = C \rightarrow g(B) = C$.

   Since h is surjective, it is the case that $h(A) = g \circ f(A) = g(f(A)) = C$.

   Note that for any sets $Y_1, Y_2 \subset X$ and any function $m : X \longrightarrow X'$, it is the case that $Y_1 \subseteq Y_2 \rightarrow m(Y_1) \subseteq m(Y_2)$.

   Thus, since $f(A) \subseteq B$, it follows that $C = g(f(A)) \subseteq g(B)$, and also, since $g : B \longrightarrow C$, we have $g(B) \subseteq C$. Therefore, $g(B) = C$.

3. [5 p] It is always true that $\#A \geq \#B$. YES NO
If yes, prove it, if no, find a counterexample.

See counterexample in subtask 1.

4. [5 p] It is always true that $\#B \geq \#C$. YES NO
If yes, prove it, if no, find a counterexample.

To show that $\#B \geq \#C$, we must show the existence of a surjection $B \twoheadrightarrow C$. This was done in subtask 2, where $g : B \longrightarrow C$ was shown to be surjective.

4 [25 p]

Suppose a graph (V, E).

You may remember from Lab 5 that a Hamiltonian path was path in the graph in which every vertex was visited exactly once.

By contrast, an *Eulerian path* is a path that uses every *edge* exactly once. A given vertex may occur multiple times in an Eulerian path (since any number of edges can start or end at that vertex), but every edge must be used exactly once.

As with Hamiltonian paths, a given graph can have zero, one, or more Eulerian paths. This task is about **computing the set of all Eulerian paths** of the graph (V, E). We do this in two steps.

1. [15 p] First, we create a function $P : V \longrightarrow \mathcal{P}(V^*)$ that computes for every vertex $v \in V$ the set of all Eulerian paths starting at v.

   It does so using a helper function $P' : V \times V^* \times \mathcal{P}(V \times V) \longrightarrow \mathcal{P}(V^*)$ as follows:

$$
\begin{aligned}
&P : V \longrightarrow \mathcal{P}(V^*)\\
&\qquad v \mapsto P'(v, \varepsilon, E)
\end{aligned}
$$

   Define P'.

   The parameters v, p, and D mean the following:

   v : the current vertex,
   p : the path up to this point, not including v,
   D : the set of edges that haven't been used yet.

$$
P' : V \times V^* \times \mathcal{P}(V \times V) \longrightarrow \mathcal{P}(V^*)
$$

$$
v, p, D \mapsto \begin{cases} \{pv\} & \text{for } D = \emptyset \\ \bigcup\limits_{w \in D(v)} P'(w, pv, D \setminus (v, w)) & \text{for } D \neq \emptyset \end{cases}
$$

2. [5 p] Now define $Q \subseteq V^*$, the set of all Eulerian paths in the graph:

$$Q = \bigcup_{v \in V} P(v) \qquad \text{or simply} \qquad P(V)$$

3. [5 p] In order to ensure that P' terminates, we require a **well-founded strict order** $\prec$ on its domain $V \times V^* \times \mathcal{P}(V \times V)$, such that for any (v, p, D) that P' is called on, it will only ever call itself on $(v', p', D') \prec (v, p, D)$. Define such an order:

$$(v', p', D') \prec (v, p, D) \iff D' \subset D$$

Remember: A correct answer to this question must have three properties.

1. It must be a strict order on $V \times V^* \times \mathcal{P}(V \times V)$.
2. It must be well-founded, i.e. there cannot be an infinite descending chain in that order.
3. Your definition of P' must conform to it, i.e. any recursive call in it must be called on a smaller (according to the order) triple of arguments.