Exam for „Algorithms I“

WS 2024/25 / February 17, 2025

First name: ____________________

Second name: ____________________

Matriculation number: ____________________

task	**max. points**	**points achieved**
1	7	
2	5	
3	6	
4	5	
5	7	
6	6	
7	6	
Σ	42	

Important information

- Duration of the exam: **60 minutes**.
- Tools: You are allowed to use a handwritten sheet of paper (size DIN A4). Both sides of the sheet of paper can be handwritten.
- Write with an indelible pen. Do not write in red paint.
- Check the exam you have been given for completeness: **7 tasks** on 9 pages.
- Enter your name and matriculation number in the appropriate fields on each sheet.
- Write your solutions in the spaces provided. If there is not enough space in a field, use the back of the corresponding sheet and indicate this on the front. If there is still not enough space, you can ask the supervisor for additional sheets of paper.
- Please write clearly. Illegible answers are invalid.
- Any attempt to cheat will result in immediate exclusion and failure. There will be no advance warning.
- All electronic devices must be switched off before the exam or at the latest now.

Task 1. (7 Points)

- Which of the following statements hold? You do not have to prove your answers.

  (1) $n^n \in \Omega(n!)$ (2) $3^{2n} \in \mathcal{O}(2^{3n})$ (3) $\frac{1}{n}\log n \in o(1)$

Solution:

(1) correct

(2) not correct

(3) correct

- Using the Master Theorems, determine the asymptotic growth (in Θ-notation) of the functions f and g subject to the following recursions.

  (4) $f(n) = 4\,f(n/2) + n^2$ (5) $g(n) = 9\,g(n/3) + n\sqrt{n}$

Solution: Both functions can be analyzed with the Master Theorem I as our functions f and g are of the general form $t(n) = a\,t(n/b) + n^c$.

(4) Here, $a = 4$, $b = 2$ and $\log_b a = 2 = c$. Hence, $f(n) \in \Theta(n^2 \log n)$.

(5) Here, $a = 9$, $b = 3$ and $\log_b a = 2 > c = 3/2$. Hence, $g(n) \in \Theta(n^2)$.

Task 2. (5 Points)

Recall that the height of a tree is the number of edges along a longest path from the root of the tree to a leaf. Prove that, for every $h \in \mathbb{N}$, there are at least 2^h distinct binary trees of heigth h.

Solution: *Alternative 1 (assuming binary trees are ordered).* There are exactly 2^h binary trees of heigth h with one leaf. Indeed, every one of the h edges along the path from the root to the leaf can point either towards the left or right child. Since there are h such choices, there are 2^h such trees.

Alternative 2. The statement is clearly true for $h = 0$. For $h > 0$, the claim follows by induction. For one subtree of the root, choose the subtree to either be empty or to consist of a single leaf. Then choose any of the at least 2^{h-1} binary trees of height $h-1$ for the other subtree of the root. There are at least $2 \cdot 2^{h-1} = 2^h$ such choices, each leading to a distinct binary tree of heigth h.

Task 3. (6 Points)

- Write down the pseudocode for the procedure **counting-sort**.

Solution: See slides.

- Sort the following array of numbers using Radix Sort (in base 10).

$$[568, 416, 538, 857, 976, 462, 389, 543]$$

Write down the new array after each call to **counting-sort**.

Solution: After the 1. call to **counting-sort** we obtain

$$[462, 543, 416, 976, 857, 568, 538, 389].$$

After the 2. call to **counting-sort** we obtain

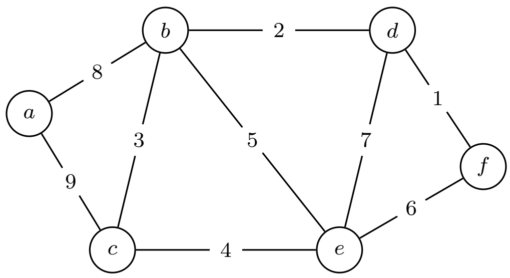
$$[416, 538, 543, 857, 462, 568, 976, 389].$$

After the 3. call to **counting-sort** we obtain

$$[389, 416, 462, 538, 543, 568, 857, 976].$$

Task 4. (5 Points)

Compute a spanning subtree of *minimal* weight using Kruskal's algorithm for the following graph. Show the edge selected in each step or indicate if no edge is selected.

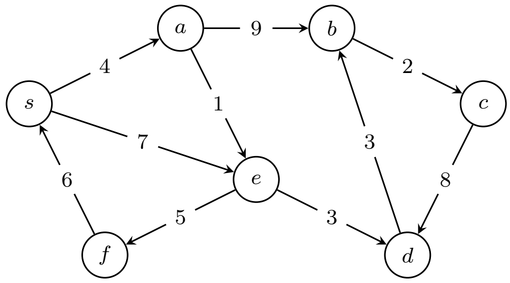


Solution: Let T be current set of selected edges. Initially we have $T = \emptyset$.

1. Select $\{d, f\}$: $T = \{\{d, f\}\}$
2. Select $\{b, d\}$: $T = \{\{b, d\}, \{d, f\}\}$
3. Select $\{b, c\}$: $T = \{\{b, c\}, \{b, d\}, \{d, f\}\}$
4. Select $\{c, e\}$: $T = \{\{b, c\}, \{b, d\}, \{c, e\}, \{d, f\}\}$
5. Don't select $\{b, e\}$.
6. Don't select $\{e, f\}$.
7. Don't select $\{d, e\}$.
8. Select $\{a, b\}$: $T = \{\{a, b\}, \{b, c\}, \{b, d\}, \{c, e\}, \{d, f\}\}$
9. Don't select $\{a, c\}$.

Task 5. (7 Points)

Use Dijkstra's algorithm to compute all shortest paths starting at node s in the graph below. Show the values of the program variables B, R, U, p, D after each iteration of the main **while**-loop of Dijkstra's algorithm.

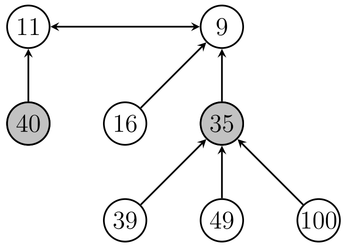


Solution: In table form: the distances are **bold** for tree nodes B, normal for boundary nodes R and ∞ for unknown nodes U. (Step 0 is optional.)

Node		s	a	b	c	d	e	f
Step 0	$D(x)$	0	∞	∞	∞	∞	∞	∞
	$p(x)$	nil	nil	nil	nil	nil	nil	nil
Step 1	$D(x)$	**0**	4	∞	∞	∞	7	∞
	$p(x)$	nil	s	nil	nil	nil	s	nil
Step 2	$D(x)$	**0**	**4**	13	∞	∞	5	∞
	$p(x)$	nil	s	a	nil	nil	a	nil
Step 3	$D(x)$	**0**	**4**	13	∞	8	**5**	10
	$p(x)$	nil	s	a	nil	e	a	e
Step 4	$D(x)$	**0**	**4**	11	∞	**8**	**5**	10
	$p(x)$	nil	s	d	nil	e	a	e
Step 5	$D(x)$	**0**	**4**	11	∞	**8**	**5**	**10**
	$p(x)$	nil	s	d	nil	e	a	e
Step 6	$D(x)$	**0**	**4**	**11**	13	**8**	**5**	**10**
	$p(x)$	nil	s	d	b	e	a	e
Step 7	$D(x)$	**0**	**4**	**11**	**13**	**8**	**5**	**10**
	$p(x)$	nil	s	d	b	e	a	e

Solution:

Task 6. (6 Points)
The following Fibonacci heap is given.

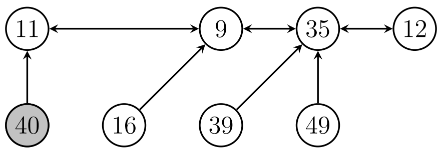


Perform the following sequence of operations on the above Fibonacci heap.

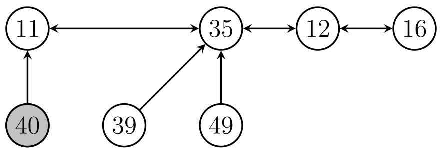
(1) **decrease-key**(node with key 100, 12)
(2) **delete-min**
(3) **delete-min**

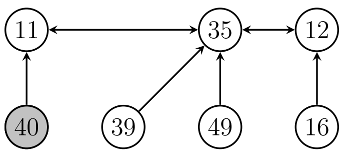
Solution:

(1) **decrease-key**(node with key 100, 12):



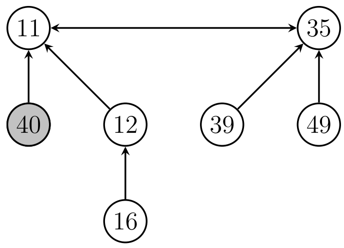
(2) **delete-min**:

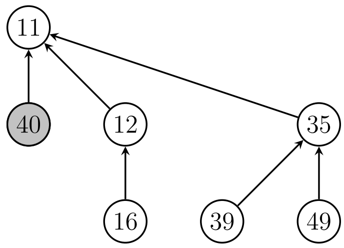




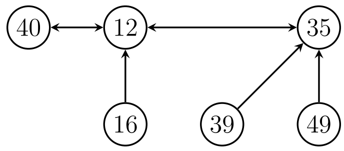
Solution:

(2) **delete-min**: (continued)





(3) **delete-min**:



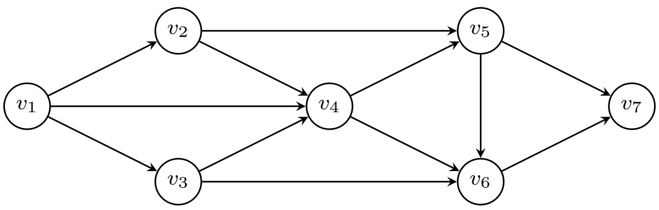
Task 7. (6 Points)

Let us consider directed graphs $G = (V, E)$ on the vertex set $V = \{v_1, v_2, \ldots, v_n\}$ and the property that $i < j$ for every edge from v_i to v_j in G. On input of such a graph G, we want to compute the number of distinct paths from v_1 to v_n.

- Give a brief description of a dynamic programming solution for this problem.

Solution: Initialize an array $P[1 \ldots n]$ with all values set to 0. (Eventually $P[i]$ will hold the number of distinct paths from v_1 to v_i.) Set $P[1] \leftarrow 1$. Then, for each edge from v_i to v_j in lexicorphic order of the pair (i, j), set $P[j] \leftarrow P[j] + P[i]$. Finally, return the value $P[n]$.

- Compute the number of distinct paths from v_1 to v_7 in the following graph.



Solution: The resulting array P is given in the follow table.

i	1	2	3	4	5	6	7
$P[i]$	1	1	1	3	4	8	12

As such, there are 12 disctinct paths from v_1 to v_7 in the given graph.