University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn

Algorithms and Data Structures Exam

8. August 2024, 14:00 -17:00

Name: ....

Matriculation No.: ....

Signature: ....

Do not open or turn until told so by the supervisor!

- Write your **name** and **matriculation number** on this page and **sign** the document.
- Your **signature** confirms that you have answered all exam questions yourself without any help, and that you have notified exam supervision of any interference.
- You are allowed to use a summary of **five handwritten, single-sided A4 pages**.
- **No electronic devices** are allowed.
- Write legibly and only use a pen (ink or ball point). Do **not use red**! Do **not use a pencil**!
- You may write your answers in **English or German** language.
- Only **one solution per task** is considered! Make sure to strike out alternative solutions, otherwise the one yielding the minimal number of points is considered.
- **Detailed steps** might help you to get more points in case your final result is incorrect.
- The keywords **Show...**, **Prove...**, **Explain...** or **Argue...** indicate that you need to prove or explain your answer carefully and in sufficient detail.
- The keywords **Give...**, **State...** or **Describe...** indicate that you need to provide an answer solving the task at hand but without proof or deep explanation (except when stated otherwise).
- You may use information given in a **Hint** without further explanation.
- **Read each task thoroughly** and make sure you understand what is expected from you.
- **Raise your hand** if you have a question regarding the formulation of a task or if you need additional sheets of paper.
- A total of **45 points** is sufficient to pass and a total of **90 points** is sufficient for the best grade.
- Write your name on **all sheets**!

Task	1	2	3	4	5	6	7	**Total**
Maximum	28	15	12	15	20	15	15	120
Points								

Task 1: Short Questions *(28 Points)*

(a) Let $A = [n, 1, 2, \ldots, n-1]$ be an array of n elements that is sorted except for the first element. What is the asymptotic runtime of Insertion Sort to sort A? What is the runtime of Merge Sort to sort A? Justify your answers. *(3 Points)*

(b) Insert the following sequence of keys into an initially empty binary search tree in the given order: 13, 5, 2, 18, 10, 11, 3, 7, 6, 4. Draw the resulting search tree. What does the search tree look like after executing delete(5)? Draw this tree as well. *(5 Points)*

(c) For a connected weighted graph $G = (V, E, w)$, we define a Maximum Spanning Tree as a spanning tree T such that its weight $w(T) = \sum_{e \in T} w(e)$ is maximized, i.e., for any other spanning tree T', we have $w(T) \geq w(T')$. Provide an efficient algorithm to compute a Maximum Spanning Tree. Prove that the algorithm is correct and analyze its runtime. *(5 Points)*

(d) Consider a weighted graph $G = (V, E, w)$ with integer, positive edge weights ($w : E \to \mathbb{N}$). We define two new weight functions:

$$w_1(e) := w(e) + 1$$
$$w_2(e) := w(e) + \frac{1}{|V|}$$

Prove or disprove the following statements:

(i) Every shortest path in G is also a shortest path in $G' = (V, E, w_1)$. *(2.5 Points)*

(ii) Every shortest path in $G' = (V, E, w_1)$ is also a shortest path in G. *(2.5 Points)*

(iii) Every shortest path in G is also a shortest path in $G'' = (V, E, w_2)$. *(2.5 Points)*

(iv) Every shortest path in $G'' = (V, E, w_2)$ is also a shortest path in G. *(2.5 Points)*

(e) To execute the Knuth-Morris-Pratt algorithm from the lecture, an array S must be precomputed. Provide this array S for the pattern $P =$`CCDCCCD`. *(5 Points)*

Solution Task 1

Task 2: Landau-Notation *(15 Points)*

(a) Sort the following functions in ascending order according to Landau notation, i.e., for any two consecutive functions f, g in this order, we have $f(n) \in O(g(n))$. No proofs or justifications are required. *(5 Points)*

- $a(n) = 3^{(n^2)} - 3n^{13}$
- $b(n) = 5^{\log_5(\log_5(n^5))}$
- $c(n) = n + |\log_5(5^{-n})|$
- $d(n) = \sqrt[800]{n^4}$
- $e(n) = (\log_2(n))^{2000}$

(b) Prove or disprove using the definition of Landau notation:

$$2^{\sqrt{\log_2 n}} \notin \Omega(100\sqrt{n})$$

(5 Points)

(c) Prove or disprove using the definition of Landau notation:

$$\log_2(n!) \in \Theta(n \log_2 n)$$

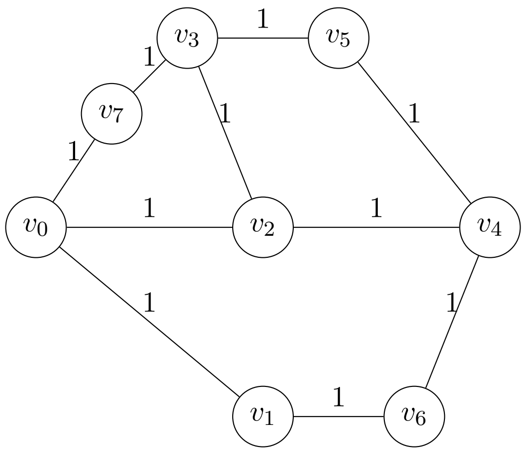
(5 Points)

Solution Task 2

Task 3: Find Nearby Nodes *(12 Points)*

The rural areas are supposed to get better access to public transport. To achieve this a new nerby-cities ticket is introduced. Using this ticket one is allowed to travel from ones town to all adjacent towns. The owners of tickets want a tool that lets them determine to which cities they are allowed to travel using their ticket. To determine the towns that can be reached using the ticket, we care only about the town of origin v and the price-level d of the ticket. The higher the price-level, the farther away towns can be reached. We model the connections between towns as a graph. Give an algorithm that determines all nodes at distance at most d from some starting node v.
You notice that no town has more than 10 neighboring towns. You want to abuse this fact to make your algorithm as fast as possible.

Formally: Given a weighted graph $G = (V, E, w)$ where all edge weights are 1 and each node has at most 10 neighbors, design an efficient algorithm to find all nodes within distance d from a given node v. Justify why the algorithm is correct and analyze its runtime in terms of $n = |V|$ and d.



In the example above, starting from node v_5 and with distance $d = 2$, the nodes $\{v_7, v_3, v_2, v_4, v_6\}$ can be reached.
Argue why your algorithm is correct and analyze the runningtime as a function of the parameters $n := |V|$ and d.
Consider the following questions and answer them explicitly:

- In what form must the graph be given to achieve the claimed runtime?
- Where in your analysis is the fact that each node has at most 10 neighbors used?
- For which values of d does your algorithm run in time $o(n)$?
  Note that this already implies that you are not even allowed to initialise an array of size n.

Solution Task 3

Task 4: Heaps with small Priorities *(15 Points)*

A priority queue stores elements with priorities between 1 and C.

(a) Design a data structure supporting insert in $O(1)$ time and delete-min in $O(C)$ time, using at most $O(n + C)$ space. Prove that your data structure satisfies these constraints. (10 Points)

(b) Modify your data structure so that delete-min runs in $O(\sqrt{C})$ time while preserving the constraints from (a). Prove that your data structure satisfies these constraints. (5 Points)

Solution Task 4

Task 5: Sweatword Filter ***(20 Points)***

Solution Task 5

Task 6: MST in partitioned Graphs ***(15 Points)***

Solution Task 6

Task 7: Dynamische Programmierung *(15 Points)*

Beim Speichern von deinem Textdokument ist ein Fehler aufgetreten! Alle Satzzeichen wie Punkt, Komma oder Leerzeichen sind verschwunden. Ein Satz aus dem Dokument würde also beipielsweise so aussehen:
"IchliebeesKlausurenzuschreiben"
Wir haben den Duden als Hashtabelle *dict* abgespeichert, in dieser können wir für jeden String w checken ob w ein gültiges Wort ist. Wenn w im Duden ist, dann gibt $dict.contains(w)$ in konstanter Zeit *True* zurück.
Beschreiben Sie einen effizienten Algorithmus der entscheidet ob ein gegebener n Buchstaben langer String $s[1, \ldots, n]$, wieder zu einem sinnvollen Text zusammengesetzt werden kann (also ob es eine Aneinanderreihung von Wörtern ist). Argumentieren Sie warum Ihr Algorithmus korrekt ist und analysieren Sie die Laufzeit.

Solution Task 7