Gedächtnisprotokoll
Algorithmen

bei Prof. Kaufmann

4. März 2021

```
Klausur:        Hauptklausur, WS, 2021
Pruefer:        Prof. Kaufmann
Datum:          03.03.2021
Zeit:           90 min + 5 min fuer Zwischenfragen
Punkte:         30
Hilfsmittel:    zwei Seiten handschriftliche Formelsammlung
Sprache:        Deutsch
Modul:          INF2420
```

Eine Aufgabe muss gestrichen werden.

1 Rekursionsgleichungen - 6 Punkte

Gegeben ist die Rekursionsgleichung:

$$T(n) = \begin{cases} \sqrt{2} \cdot T(\frac{n}{2}) + \sqrt{n} & \text{falls } n > 1 \\ 1 & \text{sonst} \end{cases}$$

Es kann angenommen werden, dass $n = 2^k,\ k \in \mathbb{N}$ eine Zweierpotenz ist.

(a) Wende das Mastertheorem auf $T(n)$ an.

(b) Finde für $T(n)$ eine geschlossene Form und beweise die Korrektheit mit vollständiger Induktion.

2 Multiple-Choice-Fragen - 6 Punkte

Kreuze eine von 4 Antworten an.

1) Gegeben sei ein vollständiger Graph $G = (V, E)$ ohne Selbstschleifen mit $|V| = n$ Knoten. Was ist die Anzahl der Kanten $|E|$?

- ☐ $\frac{n(n-1)}{2}$
- ☐ n^2
- ☐
- ☐

2) Gegeben sei ein ungerichteter, bipartiter Graph $G = (A \cup B, E)$. Was ist eine scharfe obere Schranke für die Anzahl der Kanten $|E|$?

- ☐ $|A| + |B|$
- ☐ $|A| \cdot |B|$
- ☐ $\frac{n(n-1)}{2}$, wobei $n = |A| + |B|$
- ☐ $\frac{n(n-1)}{4}$, wobei $n = |A| + |B|$

3) Sei S die Menge der Zusammenhangskomponenten eines ungerichteten Graphen $G = (V, E)$ mit $|V| = n$ Knoten und $|E| = m$ Kanten. Welche Aussage ist für einen allgemeinen Graphen G korrekt?

- ☐ Richtet man alle Kanten **beliebig**, enstspricht S den starken Zusammenhangskomponenten.
- ☐ Es ist **immer** möglich, die Kanten so zu richten, sodass die starken Zusammenhangskomponenten S entsprechen.
- ☐ Es ist **nicht immer** möglich, die Kanten so zu richten, sodass die starken Zusammenhangskomponenten S entsprechen.
- ☐ Es ist **nie** möglich, die Kanten so zu richten, sodass die starken Zusammenhangskomponenten S entsprechen.

4) Gegeben sei ein schleifenfreier, ungerichteter Graph $G = (V, E)$ mit $|V| = n$ Knoten und $|E| = m$ Kanten. Welche Antwort gibt eine scharfe Laufzeitschranke für die Berechnung aller Zusammenhangskomponenten an?

- ☐ $\mathcal{O}(n)$
- ☐ $\mathcal{O}(m)$
- ☐ $\mathcal{O}(n + m)$
- ☐ $\mathcal{O}(\log n + m)$

5) Aus welchem Grund ist die Adjazenlistendarstellung oft der Adjazenzmatrixdarstellung vorzuziehen?

- ☐ Speicherplatz von Anzahl Kanten abhängig, deshalb für Graphen mit wenigen Kanten besser.
- ☐ Einfügen und Löschen ist asymptotisch schneller.
- ☐ Matrixdarstellung geht nur für gerichtete Graphen.
- ☐ Es kann schneller festgestellt werden, dass eine Kante nicht existiert.

6) Gegeben sei ein ungerichteter Graph in Adjazenzlistendarstellung. Die Nachbarn eines Knotens werden nicht als Liste, sondern als AVL-Baum gespeichert. Was ist die Worst-Case-Laufzeit für das Einfügen einer Kante und das Aufzählen aller Nachbarn?

- ☐ $\mathcal{O}(\log n)$ und $\mathcal{O}(\log n)$
- ☐ $\mathcal{O}(n)$ und $\mathcal{O}(\log n)$
- ☐ $\mathcal{O}(\log n)$ und $\mathcal{O}(n)$
- ☐ $\mathcal{O}(\log n)$ und $\mathcal{O}(n \log n)$

(bin nicht ganz sicher, ob alle Antwortmöglichkeiten korrekt wiedergegeben sind)

3 Dijkstra-Algorithmus - 6 Punkte

(gerichteter Graph und leere Tabelle gegeben)

(a) Führe den Dijkstra-Algorithmus für den gegebenen Graphen durch und trage die Werte für jeden Zwischenschritt in die Tabelle ein.

(b) Sei G ein (beliebiger) gerichteter Graph mit Kosten $c : E \to \mathbb{N}$ und p kürzester Pfad in G von s nach t. Zeige oder widerlege: Wird zu allen Kantenkosten 7 addiert, so ist p immer noch der kürzeste Pfad von s nach t.

4 Manhattan Minimal Spanning Tree (MMST) - 6 Punkte

(Definition, Erklärungen und Beispiel für MMST)

(a) Zeichne MMST in gegebenen Graphen ein.

(b) Sei (p, q) Kante von MMST(P). Beweise, dass sich in dem von p und q aufgespannten Rechteck kein Knoten $r \notin \{p, q\}$ befindet.

5 Merge Sort - 6 Punkte

(a) Beweise, dass Merge Sort ein Array der Länge n in Zeit $\mathcal{O}(n \log n)$ sortiert.

(Erklärung zur Parallelisierung, vgl. Probeklausur)

(b) Beweise, dass parallelisiertes Merge Sort ein Array der Länge n in Zeit $\mathcal{O}(n)$ sortiert.

6 Dynamische Programmierung - 6 Punkte

Rucksackproblem: Jedes Objekt o_j hat ein Gewicht w_j. Der Rucksack soll mit Objekten gefüllt werden, sodass er so schwer wie möglich ist, allerdings darf ein Maximalgewicht W nicht überschritten werden. (genauere Beschreibung des Problems)

(a) Beispiel $W = 12$. Fülle Tabelle aus.

(b) Gib einen Algorithmus in Pseudo-Code an, der das Problem allgemein löst. Verwende **dynamische Programmierung**.

(c) Begründe die Korrektheit und Laufzeit.