Karlsruher Institut für Technologie

Institut für Theoretische Informatik
T.T.-Prof. Dr. Thomas Bläsius
https://scale.iti.kit.edu

Übungsblatt 13

Algorithmen I – Sommersemester 2022

Abgabe im ILIAS bis 27.07.2022, 14:00 Uhr

Bitte beschrifte Deine Abgabe gut sichtbar mit Deinem Namen und Deiner Matrikelnummer. Achte insbesondere bei handschriftlichen Abgaben auf Lesbarkeit und genügend Platz für Korrektur-Anmerkungen. Die Abgabe erfolgt über das Übungsmodul in der Gruppe Deines Tutoriums im ILIAS. Gib Deine Ausarbeitungen in *einer* PDF-Datei ab. Achte darauf, effiziente Algorithmen zu formulieren, also solche mit möglichst geringer asymptotischer Laufzeit!
Wenn du die Korrektheit eines Algorithmus begründen oder dessen Laufzeit analysieren sollst, tue dies getrennt von der Beschreibung des Algorithmus.
Wenn nicht anders spezifiziert oder aus dem Kontext ersichtlich, bezeichnen wir mit Graph einen einfachen ungerichteten Graphen.

Aufgabe 1 - KANNHIERMALJEMANDSPLITTEN? (8 Punkte)

Sei W eine Menge von Wörtern. Ein String S der Länge n ist *splittable*, wenn es Trennstellen $0 = s_0 < \cdots < s_k = n$ gibt, sodass $S[s_{i-1}, s_i) \in W$ für alle $i \in [1, k]$.

Wir betrachten nun das Problem MinSplits, wobei ein gegebener String in möglichst wenige Wörter gesplittet werden soll. Die Lösung einer MinSplit-Instanz besteht dann aus der Menge der zugehörigen Trennstellen. Wenn ein String nicht splittable ist, ist diese Menge leer.

1. Sei eine MinSplits-Instanz gegeben durch $S = \texttt{PRIMALGOALPHASEARCHER}$ und

$$W = \{\texttt{ALGO}, \texttt{ALPHA}, \texttt{ARC}, \texttt{ARCH}, \texttt{AS}, \texttt{EAR}, \texttt{GO}, \texttt{GOAL}, \texttt{HAS}, \texttt{HER}, \texttt{PHASE}, \texttt{PRIM}, \texttt{PRIMAL}, \texttt{RIM}, \texttt{SEA}, \texttt{SEARCH}, \texttt{SEARCHER}\}$$

   Gib die Lösung für diese Instanz an. (1 Punkt)

Nun wollen wir ein dynamisches Programm formulieren, was MinSplits löst.

2. Auf welche Teilprobleme kann das Problem reduziert werden und wie sehen Teillösungen für diese aus? (2 Punkte)

3. Gib die Rekurrenz an, mit der Teillösungen aus Teilaufgabe 2 zu einer neuen Teillösung kombiniert werden. (1 Punkt)

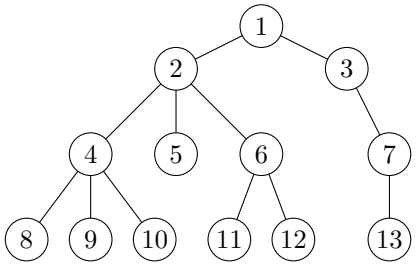
4. Beschreibe einen Algorithmus, der die Rekurrenz aus Teilaufgabe 3 verwendet, um eine Instanz von MinSplits zu lösen. Gib dazu an, wie die Teillösungen verwaltet werden, in welcher Reihenfolge die Teillösungen berechnet werden und wie man schließlich die tatsächliche Lösung erhält. Begründe warum dein Algorithmus unter der Annahme, dass das längste Wort in W aus maximal 17 Buchstaben besteht, eine asymptotische Laufzeit von $O(n \cdot |W|)$ nicht überschreitet. (3 Punkte)

5. Was ändert sich in deinem Algorithmus, wenn wir stattdessen das Problem MaxSplits lösen wollen, bei dem ein String in möglichst viele Wörter gesplittet werden soll? (1 Punkt)

Aufgabe 2 - Boston Tree Party (8 Punkte)

Sei $T = (V, E)$ ein Baum. Wir interessieren uns nun für das Problem MaximumIndependentSet, bei dem eine möglichst große Teilmenge $S \subseteq V$ gesucht wird, sodass für jede Kante $\{u, v\} \in E$ gilt: $u \notin S$ oder $v \notin S$.

1. Gib eine Lösung für folgende MaximumIndependentSet-Instanz an: (1 Punkt)



Nun wollen wir ein dynamisches Programm formulieren, was MaximumIndependentSet löst.

2. Auf welche Teilprobleme kann das Problem reduziert werden und wie sehen Teillösungen für diese aus? (2 Punkte)

3. Gib die Rekurrenz an, mit der Teillösungen aus Teilaufgabe 2 zu einer neuen Teillösung kombiniert werden. (1 Punkt)

4. Beschreibe einen Algorithmus, der die Rekurrenz aus Teilaufgabe 3 verwendet, um eine Instanz von MaximumIndependentSet zu lösen. Gib dazu an, wie die Teillösungen verwaltet werden, in welcher Reihenfolge die Teillösungen berechnet werden und wie man schließlich die tatsächliche Lösung erhält. Begründe warum dein Algorithmus eine asymptotische Laufzeit von $O(n)$ nicht überschreitet. (4 Punkte)

Aufgabe 3 - Kruskal Recall (4 Punkte)

Wir haben in der Vorlesung die Union-Find Datenstruktur kennengelernt, um Kruskal's Algorithmus effizient umsetzen zu können. Gib Kruskal's Algorithmus unter Verwendung der Union-Datenstruktur in Pseudocode an. Nenne und begründe das asymptotische Laufzeitverhalten. Verwende die folgende Signatur: (4 Punkte)

KRUSKAL$(G = (V, E) : \text{Graph}) : \text{List}\langle\text{Edge}\rangle$
