

Übungsblatt 10

Aufgabe 10.1:

(6 Punkte)

Wir betrachten die erste *Union-Find*-Datenstruktur aus der Vorlesung, d. h. ein n -elementiges Feld A , in dem für jedes Element $i \in \{1, \dots, n\}$ gespeichert ist, zu welcher Menge es gehört. Zusätzlich ist für jede Menge ihre Größe und eine Liste der Elemente der Menge gegeben.

In der Vorlesung wurde gezeigt, dass die Laufzeit für jede Folge von UNION-Operationen auf n Elementen höchstens $O(n \log n)$ ist. Wie sieht eine Folge solcher UNION-Operationen auf n Elementen aus, deren Laufzeit wirklich $\Theta(n \log n)$ beträgt? Beweisen Sie Ihre Aussage. Beschränken Sie sich dabei auf den Fall, dass n eine Zweierpotenz ist.

Aufgabe 10.2:

(6 Punkte)

Implementieren Sie einen Algorithmus, der für einen gegebenen ungerichteten Graphen $G = (V, E)$ die Anzahl der Zusammenhangskomponenten und deren Größen ausgibt. Wenden Sie Ihren Algorithmus auf die Instanzen

- (a) `Ue10_small.txt` mit $|V| = 50$,
- (b) `Ue10_medium.txt` mit $|V| = 200$ und
- (c) `Ue10_large.txt` mit $|V| = 500$

an, die auf der Homepage heruntergeladen werden können. Das Dateiformat ist dort ebenfalls erklärt.

Aufgabe 10.3:

(3+3 Punkte)

Es sei A die Adjazenzmatrix eines (gerichteten oder ungerichteten) Graphen. Das Boolesche Matrixprodukt $Z = X * Y$ zweier $(n \times n)$ -Matrizen X und Y über $\{0, 1\}$ ist definiert durch

$$z_{ij} = \bigvee_{\ell=1}^n (x_{i\ell} \wedge y_{\ell j}).$$

Dabei werden, wie üblich, 1 als *true* und 0 als *false* interpretiert. Es sei A^k die k -te Potenz von A , d. h. $A^1 = A$ und $A^k = A * A^{k-1}$ für $k \geq 2$.

- (a) Welche Bedeutung haben die Einträge von A^k ? Beweisen Sie Ihre Vermutung!
- (b) Sei $f(n)$ der Aufwand, zwei gegebene $(n \times n)$ -Matrizen X und Y über $\{0, 1\}$ zu multiplizieren. Beschreiben Sie, wie man für $k \geq 2$ in Zeit $O(\log(k) \cdot f(n))$ die k -te Potenz A^k einer Adjazenzmatrix A berechnen kann. Sie können davon ausgehen, dass das Boolesche Matrixprodukt assoziativ ist. Das bedeutet insbesondere, dass $A^{k_1} * A^{k_2} = A^{k_1+k_2}$ für beliebige natürliche Zahlen k_1 und k_2 gilt.

Hinweis: Überlegen Sie sich zunächst, wie man mit möglichst wenigen Multiplikationen spezielle große Potenzen von A berechnet. Nutzen Sie diese Idee, um A^k für beliebige Exponenten k schnell zu bestimmen.

Aufgabe 10.4:

(6 Punkte)

Gegeben sei ein minimaler Spannbaum T eines gewichteten Graphen $G = (V, E)$ als verkettete Liste von Kanten. Wir reduzieren das Gewicht einer Kante $e \in E$ von $w(e)$ auf $w'(e)$. Geben Sie einen Algorithmus an, der als Eingabe den Graphen G , den Spannbaum T und die Kante e erhält und in Zeit $O(|V|)$ einen minimalen Spannbaum für den Graphen G mit der modifizierten Gewichtsfunktion berechnet.

Aufgabe 10.5:

(6 Zusatzpunkte)

Wir wollen für ein Feld A mit n paarweise verschiedenen Einträgen drei Funktionen `INIT`, `PrintSmallerValues` und `PrintSmallest` bereitstellen. Die Funktion `INIT` soll einmal am Anfang aufgerufen werden und dient als Vorverarbeitungsschritt. Das Ausführen soll in Zeit $O(n)$ möglich sein. Ein Aufruf `PrintSmallerValues(z)` soll in Zeit $O(k)$ alle Einträge von A ausgeben, die kleiner oder gleich z sind. Die Zahl k sei dabei die Anzahl dieser Einträge (oder 1, falls alle Einträge größer als z sind). Ein Aufruf `PrintSmallest(k)` für $1 \leq k \leq n$ soll in Zeit $O(k)$ die k kleinsten Einträge von A ausgeben.

Stünde der `INIT`-Funktion Zeit $O(n \log n)$ zur Verfügung, dann könnte man das Feld sortieren.

In `PrintSmallerValues` und `PrintSmallest` würde man dann das sortierte Feld von links nach rechts durchgehen und an geeigneter Stelle stoppen. Da der `INIT`-Funktion nur Zeit $O(n)$ zur Verfügung steht ist es also nicht möglich das Feld komplett zu sortieren.

Geben sie eine `INIT`-, `PrintSmallerValues`- und `PrintSmallest`-Funktion an. Begründen Sie die Korrektheit Ihrer Funktionen und erklären Sie, warum die Laufzeitrestriktionen eingehalten werden.

Hinweis: Nutzen Sie die Tatsache, dass das Bestimmen des k -t kleinsten Eintrages eines Feldes in Linearzeit (in der Feldlänge) möglich ist, auch wenn k Teil der Eingabe ist. (siehe Aufgabe 2.4)