Matthias Krause
Alexander Moch

Mannheim, den 16.11.2020

CS 307 Algorithmen und Datenstrukturen, Herbstsemester 2020

Lösungsskizze zum Übungsblatt 8

AUFGABE 8.1$^{\mathbf{K}}$:

Auf einer UNION-FIND-Datenstruktur mit verketteten Listen (Linked Lists) sollen folgende Operationen durchgeführt werden:

`MAKE-SET(1)`, `MAKE-SET(2)`, ..., `MAKE-SET(10)`,
`UNION(1,2), UNION(3,4), UNION(2,5), UNION(6,7), UNION(4,8), UNION(9,10),`
`UNION(1,10), UNION(3,10), UNION(7,8).`

Hinweis: Falls die Größe der Listen nicht ausschlaggebend ist, so soll gelten: Die Ausführung von `UNION(i,j)` ändert den Wert von `FIND(i)` nicht. (SimpleUnion)

a) Wie sieht die verkettete Liste aus, wenn alle `UNION`s ohne Berücksichtigung der Größe der Listen ausgeführt wurden? Wie oft mussten dabei Repräsentantenzeiger geändert werden?

Wir starten mit den disjunkten Listen $(1), (2), \ldots, (10)$ *und erhalten als Zwischenschritte (nach jeder Union-Operation, Repräsentant fett gedruckt, Repräsentantenzeigeränderungen unterstrichen)):*

1. $(\mathbf{1}), (\mathbf{2}), (\mathbf{3}), (\mathbf{4}), (\mathbf{5}), (\mathbf{6}), (\mathbf{7}), (\mathbf{8}), (\mathbf{9}), (\mathbf{10})$
2. $(\mathbf{1}, \underline{2}), (\mathbf{3}), (\mathbf{4}), (\mathbf{5}), (\mathbf{6}), (\mathbf{7}), (\mathbf{8}), (\mathbf{9}), (\mathbf{10})$ $\quad \underline{1}$
3. $(\mathbf{1}, 2), (\mathbf{3}, \underline{4}), (\mathbf{5}), (\mathbf{6}), (\mathbf{7}), (\mathbf{8}), (\mathbf{9}), (\mathbf{10})$ $\quad \underline{1}$
4. $(\mathbf{1}, 2, \underline{5}), (\mathbf{3}, 4), (\mathbf{6}), (\mathbf{7}), (\mathbf{8}), (\mathbf{9}), (\mathbf{10})$ $\quad \underline{1}$
5. $(\mathbf{1}, 2, 5), (\mathbf{3}, 4), (\mathbf{6}, \underline{7}), (\mathbf{8}), (\mathbf{9}), (\mathbf{10})$ $\quad \underline{1}$
6. $(\mathbf{1}, 2, 5), (\mathbf{3}, 4, \underline{8}), (\mathbf{6}, 7), (\mathbf{9}), (\mathbf{10})$ $\quad \underline{1}$
7. $(\mathbf{1}, 2, 5), (\mathbf{3}, 4, 8), (\mathbf{6}, 7), (\mathbf{9}, \underline{10})$ $\quad \underline{1}$
8. $(\mathbf{1}, 2, 5, \underline{9}, \underline{10}), (\mathbf{3}, 4, 8), (\mathbf{6}, 7)$ $\quad \underline{2}$
9. $(\mathbf{3}, 4, 8, \underline{1}, \underline{2}, \underline{5}, \underline{9}, \underline{10}), (\mathbf{6}, 7)$ $\quad \underline{5}$
10. $(\mathbf{6}, 7, \underline{3}, \underline{4}, \underline{8}, \underline{1}, \underline{2}, \underline{5}, \underline{9}, \underline{10})$ $\quad \underline{8}$

Also lautet die endgültige Liste:

$6 \leftarrow 7 \leftarrow 3 \leftarrow 4 \leftarrow 8 \leftarrow 1 \leftarrow 2 \leftarrow 5 \leftarrow 9 \leftarrow 10$
Es wurden $1+1+1+1+1+1+2+5+8 = 21$ *Repräsentantenzeiger geändert.*

b) Wie sieht die verkettete Liste aus, wenn alle `UNION`s als *union by rank* ausgeführt wurden? Wie oft mussten dabei Repräsentantenzeiger geändert werden?

Schritte $1.-8.$ *sind identisch zu Teil a), erst ab* ***`UNION(3,10)`*** *unterscheiden sich die Lösungen:*

9. $(\mathbf{1}, 2, 5, 9, 10, \underline{3}, \underline{4}, \underline{8}), (\mathbf{6}, 7)$ $\quad \underline{3}$
10. $(\mathbf{1}, 2, 5, 9, 10, 3, 4, 8, \underline{6}, \underline{7})$ $\quad \underline{2}$

Also lautet die endgültige Liste:

$1 \leftarrow 2 \leftarrow 5 \leftarrow 9 \leftarrow 10 \leftarrow 3 \leftarrow 4 \leftarrow 8 \leftarrow 6 \leftarrow 7$
Es wurden $1+1+1+1+1+1+2+3+2 = 13$ *Repräsentantenzeiger geändert.*

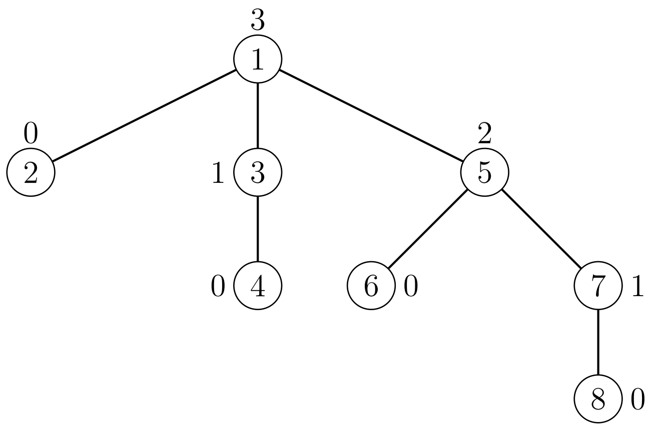
c) Angenommen, es darf nach den angegebenen `MAKE-SET`s eine beliebige Folge von `UNION`s gewählt werden. Wie oft müssen dabei mindestens Repräsentantenzeiger geändert werden, bis nur noch eine einzige Menge übriggeblieben ist? Geben Sie eine solche `UNION`-Folge an.

Es müssen mindestens 9 Repräsentantenzeiger geändert werden.
`UNION(1,2), UNION(1,3), ..., UNION(1,10)`

AUFGABE 8.2^K:

In dieser Aufgabe betrachten wir UNION-FIND-Datenstrukturen basierend auf disjunkten Bäumen mit *union by rank*. Führen Sie die folgenden Operationen durch und geben Sie den am Ende resultierenden Baum an. Geben Sie auch den Rang `rank(`$[x]$`)` für jeden Knoten x an.

`MAKE-SET(1), MAKE-SET(2),...,MAKE-SET(8),`
`UNION(1,2), UNION(3,4), UNION(5,6), UNION(7,8), UNION(1,3), UNION(5,7), UNION(1,5)`



AUFGABE 8.3:

Wir betrachten UNION-FIND-Datenstrukturen mit verketteten Listen und *union by rank*. Angenommen, es sind zuerst die Operationen MAKE-SET$(x_1), \ldots,$ MAKE-SET(x_n) ausgeführt worden. Nehmen Sie vereinfachend an, dass $n = 2^k$ für ein $k \in \mathbb{N}$.
Finden Sie eine Folge von weniger als n UNION-Operationen, die Zeit $\Omega(n \log(n))$ benötigt. Den Aufwand für eine UNION-Operation können Sie durch die Größe der kleineren der beiden beteiligten Mengen abschätzen.

Vereinige im s-ten Schritt, $s = 1 \ldots k$, die $n/2^{s-1}$ Mengen der Größe 2^{s-1} zu $n/2^s$ Mengen der Größe 2^s (Aufwand: $\frac{1}{2}\frac{n}{2^{s-1}}2^{s-1} = \frac{n}{2}$). Der Gesamtaufwand ist also $k\frac{n}{2} \in \Theta(n\log(n))$.
(Nach genau $n-1$ UNIONs bleibt eine einzige Menge übrig.)

AUFGABE 8.4$^{\mathbf{K}}$:

Wir betrachten den ungerichteten Graphen $G = (V, E)$, gegeben durch $V = \{a, b, c, d, e, f\}$ und

$$E = \{(a, c), (a, f), (b, d), (b, e), (c, f), (d, e)\}\,.$$

Berechnen Sie die Zusammenhangskomponenten von G mittels des aus der Vorlesung bekannten Algorithmus *ConnectedComponents* (Folie 166):

```
FORALL v in V DO
  MakeSet(v)
FORALL e = (u,v) in E DO
  IF Find(u) != Find(v) THEN
    Union(u,v)
```

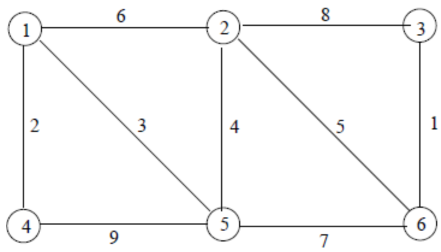
Ergebnis der ersten Schleife (Initialisierung):
$\rightarrow \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}\}$

(Zwischen-)Ergebnisse der zweiten Schleife:
$(a, c) \rightarrow \{\{a, c\}, \{b\}, \{d\}, \{e\}, \{f\}\}$
$(a, f) \rightarrow \{\{a, c, f\}, \{b\}, \{d\}, \{e\}\}$
$(b, d) \rightarrow \{\{a, c, f\}, \{b, d\}, \{e\}\}$
$(b, e) \rightarrow \{\{a, c, f\}, \{b, d, e\}\}$
$(c, f) \rightarrow \{\{a, c, f\}, \{b, d, e\}\}$ *(unverändert)*
$(d, e) \rightarrow \{\{a, c, f\}, \{b, d, e\}\}$ *(unverändert)*

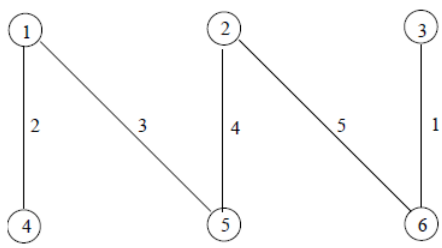
Resultat:
G hat die beiden Zusammenhangskomponenten $\{a, c, f\}$ und $\{b, d, e\}$.

AUFGABE 8.5$^{\mathbf{K}}$:

Gegeben sei der folgende ungerichtete, gewichtete Graph:



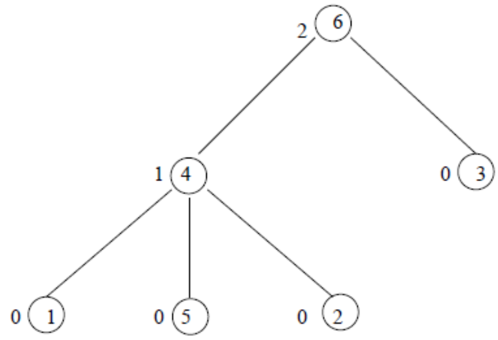
a) Benutzen Sie den Algorithmus von Kruskal, um einen minimalen Spannbaum für diesen Graphen zu ermitteln. Wie sieht der so bestimmte minimale Spannbaum aus?



b) Gehen Sie davon aus, dass bei der Berechnung des minimalen Spannbaums in der Teilaufgabe (a) eine UNION-FIND-Datenstruktur zur Verwaltung der Zusammenhangskomponenten verwendet wurde. Zu Beginn des Algorithmus wurden also die Befehle `MAKE-SET(1),...,MAKE-SET(6)` ausgeführt. Welche Abfolge von `UNION`-Anweisungen wurde danach durch den Algorithmus von Kruskal ausgeführt?

`UNION`$(3,6)$, `UNION`$(1,4)$, `UNION`$(1,5)$, `UNION`$(2,5)$, `UNION`$(2,6)$

c) Nehmen Sie nun weiterhin an, dass zur Realisierung der UNION-FIND-Datenstruktur in der Teilaufgabe (b) disjunkte Bäume verwendet wurden. Wie sieht nach Beendigung der `UNION`-Operationen aus (b) der Baum aus, in dem die Knoten verwaltet werden? Geben Sie auch den Rang `rank[`x`]` für jeden Knoten x an! [1]



[1] Zur Eindeutigkeit nehmen wir an, dass für jede Kante (u, v) gilt $u < v$. Ausserdem nehmen wir an, dass bei Gleichheit der `rank` vom zweiten Argument (bei (x, y) also `rank[`y`]`) verändert wird.