Algorithmen und Berechnungskomplexität I, WS 12/13
Aufgabenblatt 7
Universität Bonn, Institut für Informatik, Abteilung I

- *Die Lösungen können bis Dienstag, 04.12., 12:15 Uhr in den Postkasten im AVZ III eingeworfen werden.*

Aufgabe 25: Union-Find (4 Punkte)

Betrachten Sie eine Union-Find-Datenstruktur, die durch verkettete Listen repräsentiert wird und zu Anfang leer ist. Wie sieht die Union-Find-Datenstruktur aus, die durch Anwendung der folgenden Folge von Anweisungen entsteht? Wir erlauben dabei, dass eine Menge nicht nur durch ihren Repräsentanten angesprochen wird, sondern alternativ durch einen explizit vergebenen Bezeichner.

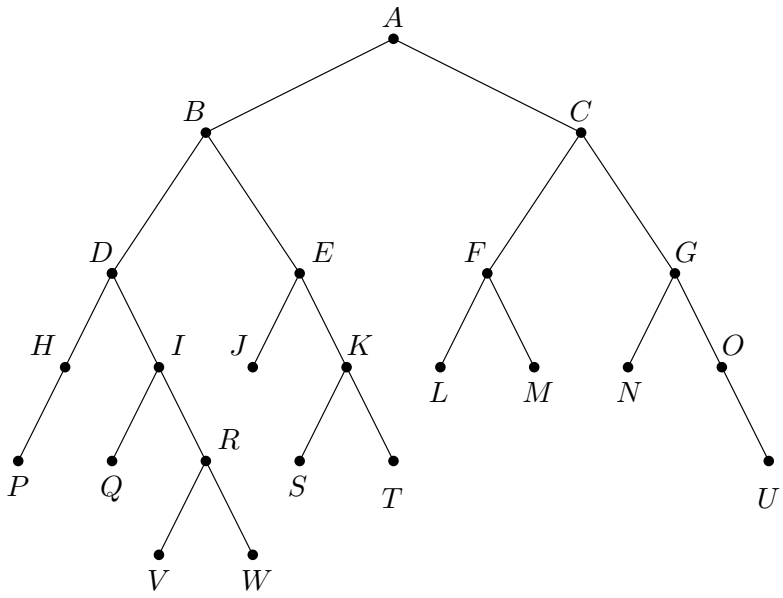
1. Make-Set(1)
2. Make-Set(2)
3. Make-Set(3)
4. Make-Set(4)
5. Make-Set(5)
6. Make-Set(6)
7. A := Union(1,2)
8. B := Union(3,4)
9. C := Union(B,6)
10. D := Union(A,C)

Geben Sie auch die Zwischenergebnisse nach den Schritten 7 bis 9 an!

Bitte wenden!

Aufgabe 26: *AVL*-Baum: Elemente löschen (4 Punkte)

Löschen Sie aus dem abgebildeten *AVL*-Baum den Knoten mit Schlüssel *C*. Beschreiben Sie wie der Knoten entfernt wird und welche Schritte anschließend zum Rebalancieren durchgeführt werden.



Aufgabe 27: Breitensuche (4 Punkte)

Gegeben einen zusammenhängenden, ungerichteten Graphen $G = (V, E)$ mit Knotenmenge $V = \{1, \ldots, n\}$ und Kantenmenge E: beschreiben Sie in *Pseudocode* einen Algorithmus, der G mit Breitensuche von Startknoten $v = 1$ aus durchsucht. Unterscheiden Sie dabei die folgenden zwei Fälle:

- Der Graph G ist in Form einer Adjazenzmatrix gegeben, also einer $n \times n$-Matrix $A[.][.]$, wobei
$$A[i][j] = 1 \Leftrightarrow (i, j) \in V$$
und
$$A[i][j] = 0 \Leftrightarrow (i, j) \notin V$$
gilt.
- G ist in Form von Adjazenzlisten gegeben, d.h. für jeden Knoten $i \in V$ gibt es eine Liste L_i, die für jeden in G zu Knoten i benachbarten Knoten j einen Container mit Inhalt j enthält.

Geben Sie in beiden Fällen die Laufzeit Ihres Algorithmus in Θ-Notation an.

Aufgabe 28: Hashing (4 Punkte)

Erstellen Sie durch sukzessives Einfügen eine Hashtabelle mit $m = 11$ Zellen für die Schlüsselwerte
$$10, 22, 31, 4, 15$$

a) Verwenden Sie Kollisionsbehandlung mittels verketteter Listen mit der primären Hashfunktion $h(x) = x \mod 7$.

b) Verwenden Sie lineare Sondierung mit der primären Hashfunktion $h(x) = x \mod 7$.

Beim der linearen Sondierung wird im Falle einer Kollision das nächste freie Feld in der Hashtabelle benutzt. Für ein Element x und eine Hashtabelle mit m Feldern wird somit folgende Probefolge erzeugt:
$$g(x, i) = (h(x) + i) \mod m.$$
Dabei wird solange iteriert, bis ein Wert für i gefunden wurde, der zu einem noch nicht belegten Tabellenfeld führt.