



# CHRISTIAN-ALBRECHTS-UNIVERSITÄT ZU KIEL

Institut für Informatik, Arbeitsgruppe Algorithmen und Komplexität  
Prof. Dr. K. Jansen, K.-M. Klein, F. Land M. Rau

09. Juni 2016

## Präsenzaufgaben zur Vorlesung »Algorithmen und Datenstrukturen«

### Blatt 9

#### Präsenzaufgabe 9.1 (Union-Find)

Betrachten Sie den dritten Ansatz zu Union Find mit dem Universum  $\{1, \dots, 7\}$ . Vereinigen Sie die Mengen in folgender Reihenfolge:

*Vereinigung*(1,2), *Vereinigung*(3,4), *Vereinigung*(5,1), *Vereinigung*(6,7), *Vereinigung*(3,6), *Vereinigung*(1,3).

Geben Sie die entsprechenden Baumstrukturen nach jeder Operation an. Wobei Vereinigung mit Ausgleich verwendet wird.

#### Präsenzaufgabe 9.2 (Union-Find)

Betrachten Sie den dritten Ansatz zu Union Find. Zeigen Sie, dass zwei verschiedene Knoten gleichen Ranges disjunkte Unterknotenmengen haben.

#### Präsenzaufgabe 9.3 (Union-Find)

Entwerfen Sie ein Union-Find Datenstruktur, die die Mengen als Listen verwaltet. Jedes Listenelement hat einen Zeiger auf das erste Element der Liste und einen Zeiger auf das nächste Element in der Liste. Bei der Vereinigung wird die kürzere Liste an die längere Liste angehängt.

Überlegen Sie sich, wie diese Datenstruktur in Java mittels eines Arrays realisiert werden kann. Implementieren Sie die Funktionen *Vereinigung* und *Finde* sowie den Konstruktor *UnionFind*.

```
class UnionFind{

    private int[][] sets;

    private static final int SIZE = 0;
    private static final int NEXT = 1;
    private static final int FIRST = 2;
    private static final int LAST = 3;
    private static final int NONE = -1;

    public UnionFind(int m){}

    public int Finde(int i){}

    public void Vereinigung(int i, int j){}

}
```

Überlegen Sie sich wie groß der Aufwand von  $m$  Finde- und Suchoperationen über eine  $n$ -elementige Grundmenge im Worst-Case sein kann, falls nicht gewährleistet wird, dass die kürzere Liste an die längere Liste gehängt wird. Geben Sie ein Beispiel an, in dem diese Laufzeit erreicht wird.