

Tutorium 10

Algorithmen I SS 14

Institut für Theoretische Informatik



Union-Find ist eine Partition einer Menge so, dass folgende Operationen effizient durchführbar sind:

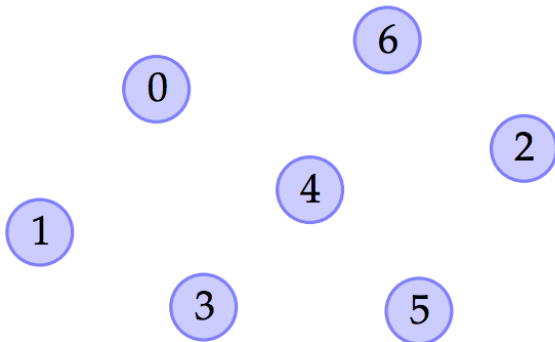
$\text{union}(x,y)$ Vereinigt die Menge mit x mit der Menge mit y

$\text{find}(x)$ Gibt den Repräsentanten der Menge mit x zurück

Herausfinden ob x in der selben Menge wie y : $\text{find}(x) == \text{find}(y)$

Repräsentation der Mengen als Bäume: Repräsentanten sind Wurzel

Zu Beginn: Die Partition besteht aus n ein-elementigen Mengen



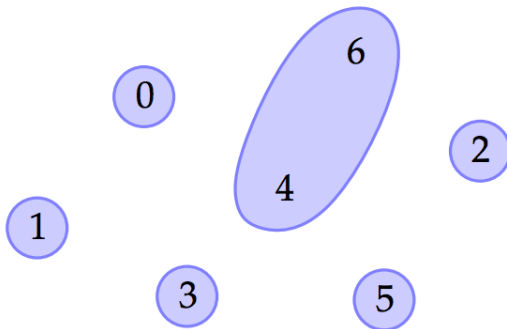
Beispiel Union-Find

`union(4,6)`

`find(4) == find(6): true`

`find(1) == find(3): false`

`find(5) == find(2): false`



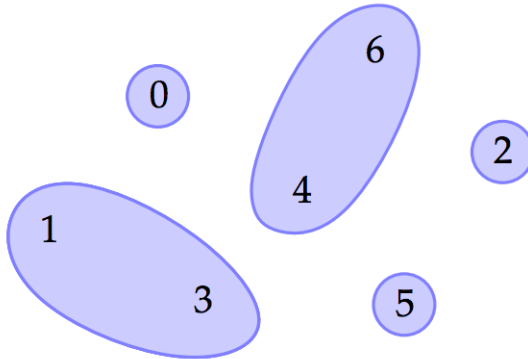
Beispiel Union-Find

`union(1,3)`

`find(4) == find(6): true`

`find(1) == find(3): true`

`find(5) == find(2): false`



`find()` Alle traversierte Knoten direkt an die Wurzel gehängt:
→ Reduziert Baumhöhe.

`union()` Der kleine Baum wird an den Größeren gehängt

- Amortisierte Laufzeit pro Operation: $\mathcal{O}(\alpha(n))$
 - $\alpha(n)$ ist die Inverse Ackermannfunktion
 - sehr langsam wachsend:

$$\alpha(2^{2^{10^{19792}}}) < 5$$