

Union-Find

1 Beschreibung

Es soll eine Menge von Mengen verwaltet werden, wobei jede Menge einen „Namen“ trägt. Oftmals wird ein kanonisches Element ausgewählt, welches den Namen der Menge bestimmt. Weiter sollen drei Operationen möglich sein:

1. $\text{makeSet}(a)$: Erzeugt eine Menge mit dem Element a .
2. $\text{find}(a)$: Findet das Element a und gibt den Namen der Menge, in welcher a vorkommt, zurück.
3. $\text{union}(M, N)$: Vereinigt die Mengen M und N .

Eine mögliche Implementierung sieht folgendermassen aus: Die Mengen werden als allgemeine Bäume gespeichert, wobei jeder Knoten einen Elternzeiger besitzt. Für die Wurzel, welches zugleich das kanonische Element darstellt, zeigt dieser auf das Element selbst. Die drei Operationen sind nun trivial zu implementieren:

1. Erzeuge ein Element mit einem Zeiger auf sich selbst.
2. Gehe zum Element a und folge den Elternzeigern, bis das kanonische Element gefunden wurde.
3. Hänge die eine Wurzel unter die Wurzel der anderen Menge.

Eine Menge zu erzeugen und zwei Mengen zu vereinigen geht in konstanter Zeit. Ein Element zu finden, geht in $O(h)$ wenn h die Höhe des Baumes bezeichnet. Will man h logarithmisch in n , der gesamten Anzahl Elemente, limitieren, so muss beim Vereinigen nicht wahllos vorgegangen werden. Entweder, man vereinigt nach Höhe, oder nach Grösse:

- **Höhe**: Der kleinere Baum (kleiner in der Höhe) wird unter den grösseren gehängt.
- **Grösse**: Der kleinere Baum (weniger Elemente) wird unter den grösseren gehängt.

Beide Verfahren liefern logarithmische Laufzeit, wobei bei beiden gewisse Informationen bei den Knoten gespeichert werden müssen (Höhe / Grösse des Teilbaumes).