

---

## Effiziente Algorithmen und Datenstrukturen I

---

*Abgabetermin: 30.01.2004 vor der Vorlesung*

### Aufgabe 1

Wir implementieren eine Union-Find-Datenstruktur wie folgt: Für die Union-Operation benutzen wir die bekannte Union-By-Size-Heuristik, für die Find-Operation die **Partial-Path-Compression**-Heuristik. Die Partial-Path-Compression-Heuristik für  $\text{FIND}(u)$  ist folgendermaßen definiert: Die Knoten entlang des Pfades von  $u$  zur Wurzel des Baumes, in dem sich  $u$  befindet, werden zu *Kindern ihrer Großeltern* gemacht (Erinnerung: Bei der ursprünglichen Path-Compression-Heuristik wurden alle Knoten entlang dieses Pfades zu *Kindern der Wurzel* gemacht). Zeigen Sie, dass die Laufzeit für eine Folge von  $n$  Union- und Find-Operationen immer noch  $O(n \log^*(n))$  ist.

### Aufgabe 2

Gegeben seien zwei sortierte Listen der Länge  $n$  bzw.  $m$ . Geben Sie einen Algorithmus an, der mit  $O(\log(n + m))$  Vergleichen den Median der beiden Listen findet.

### Aufgabe 3

Zeigen Sie, dass zum Bestimmen des größten und zweitgrößten Elementes aus einer  $n$ -elementigen Menge  $n + \lceil \log(n) \rceil - 2$  Vergleiche ausreichend sind.

### Aufgabe 4

Zeigen Sie, dass zum Bestimmen des größten und kleinsten Elementes aus einer  $n$ -elementigen Menge  $n + \lceil n/2 \rceil - 2$  Vergleiche ausreichend sind.