

# Tutorium 4

## Algorithmen I SS 14

Institut für Theoretische Informatik



Letztes Tut: Worst case  $\Leftrightarrow$  Pivot ist immer Max/Min

Gedankenspiel:

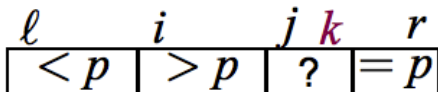
Array bestehend aus nur gleichen Elementen:

$\langle 2, 2, 2, 2 \rangle$

Wie viele Vergleiche?

→ Auf Mengen mit vielen gleichen Elementen:  
Standard-Quicksort schlecht

# Stattdessen: 3-way-partition



- 3 Pointer:  $i, j, k$
- Am Ende den  $(= p)$  Teil zwischen  $(< p)$  und  $(> p)$  schieben.

- Array aus anfänglich leeren Buckets, denen jeweils ein Schlüssel zugewiesen ist
- Basierend auf den Schlüssel werden die Elemente in die Buckets sortiert

## Eigenschaften

stabil

**Nicht inplace:** Im Worse case  $\mathcal{O}(n * k)$  Komplexität

Average Laufzeit  $\mathcal{O}(n + k)$

→ Sinnvoll bei kleiner Schlüsselmenge

Sortieren nach einzelnen Ziffern

Mehrere Varianten:

- Beginnend beim Most Significant Digit
- Beginnend beim Least Significant Digit

Eigenschaften

stabil

**Nicht inplace:** Worse case  $\Theta(d * (n + k))$  Komplexität

Laufzeit  $\mathcal{O}(d * n)$  für  $d = \text{Anzahl digits}$

→ Sinnvoll bei kleiner Wertmenge

## Pro Nicht Vergleichsbasiert:

- asymptotisch schneller

## Pro Vergleichsbasiert:

- Weniger Voraussetzungen an die zu sortierenden Elemente
- Cache-Effizienz weniger schwierig
- bei langen Schlüsseln oft schneller
- robust gegen beliebige Eingabeverteilungen