

Erste Schritte: Suche und Performance

Lernziele:

- Sie können einen Binary Search auf eine Liste anwenden
- Sie können die Laufzeit für die Suche abschätzen (Anzahl Operationen)

1 Ein einfacher Such-Algorithmus: Binary Search

Nehmen wir an, sie müssten in einem Telefonbuch nach einem Namen suchen. Wie gehen Sie vor?

Wir gehen ähnlich vor bei einem Binary Search: wir schränken den Suchraum ein, indem wir den „Raum“ halbieren.

Implementieren Sie einen Binary Search mit einem primitiven Array als fixe Grösse (Sie können auch einen Array mit dynamisch generierten Werten erstellen). Achten Sie darauf, dass Sie den Suchraum entsprechend festlegen (index des Arrays von 0 bis n-1) und diesen dann stets halbieren.

2 Anzahl Schritte

Bei einer simplen Suche sind die Anzahl Schritte im schlimmsten Fall Grösse der Liste (also bei einer Liste von 1 bis 100 wären das im schlimmsten Fall 100 Schritte).

Wenn wir mit der binären Suche arbeiten, dann ist die Anzahl Schritte deutlich kleiner und zwar \log_2 . Das würde heissen, bei einer Liste von 100 Elementen ist die Suche $\log_2 100 = \log_2(2^{\text{hoch}x} = 100)$ fertig.

Das wären... ca. 7 Schritte.

Die Anzahl Schritte sind sehr wichtig bei einem Algorithmus. Sie zeigen auf, wie schnell ein Algorithmus im schlimmsten Fall determiniert (d.h. zum Ende kommt).

Dieses Thema wird uns auch später wieder beschäftigen, wenn wir die *Performance* von Algorithmen näher betrachten.