Vorlesung Theoretische Informatik

O-Notation und Landau-Symbole

Prof. Dr. Fritz Laux

Fakultät für Informatik
Masterstudiengang Wirtschaftsinformatik
(Stand: 22.03.2010)

Sommersemester 2010

Hochschule Reutlingen
Reutlingen University

Datenstrukturen

Datenstruktur

- Array
- Hash-Table/Dictionary

Zugriffszeit

Beide Strukturen sind direkt addressierbar, d.h. die Adressen sind mit konstantem Rechenaufwand berechenbar

- Arrayadresse: <offset>+ <index>* <Bytes/Zelle>
- Hashadresse: <offset>+ <hash>* <Satzlänge>

Zugriffszeit ist konstant, d.h. unabhängig vom Index bzw. Schlüssel

Zugriffszeit = O(1)

Datenstrukturen

Datenstruktur

- Liste
- Sequentielle Datei

Zugriffszeit

Beide Strukturen werden linear addressierbar, d.h. die Elemente werden sequentiell gelesen bis zum gesuchten Element

- Durchschnitt: $n/2$ Zugriffe
- Maximal: n Zugriffe

Zugriffszeit = O(n)

Datenstrukturen

Baumstruktur

- Binärbaum: jeder Knoten hat maximal 2 Kinder
- Baum vom Grad k: jeder Knoten hat maximal k Kinder

Zugriffszeit für Bäume mit n Knoten

- Binärbaum: $O(log_2(n))$
- k-Baum: $O(log_k(n))$

Ein beliebiger Knoten ist mit maximal $log_{Grad}(n)$ Zugriffen erreichbar
Siehe Graphentheorie

Programmstrukturen

Schleifen

- einfache Schleife: n * <Rechenzeit für Schleifenrumpf>
- 2 geschachtelte Schleifen: n * m * <Rechenzeit für Schleifenrumpf>
- bei r Schleifen mit je n Durchläufen: n^r * <Rechenzeit für Schleifenrumpf>

Programmschleifen = $O(n^r)$ für r geschachtelte Schleifen

Programmstrukturen

Suche in Collections

- OrderedCollection: O(n)
- Set: O(1) für Schlüsselsuche, O(n) sonst
- Array: O(1) für Indexsuche, O(n) sonst

Einfügen/Löschen in Collections

- OrderedCollection: O(1) (O(n) bei Reorganisation)
- Set: O(1) (O(n) bei Reorganisation)
- Array: O(1) für Update