

Gliederung Kuckuckshashing

Sebastian Happel

14. Oktober 2023

1 Einleitung

In der Einleitung führe ich kurz in Hashmethoden im Zusammenhang mit Dictionaries ein und leite damit zum Kuckuckshashing über.

Zusätzlich definiere ich die Datenstruktur des Dictionary und die darauf anwendbaren Operationen.

2 Grundlagen

Hier ist das Ziel Dinge zu erklären, die später in der Ausarbeitung relevant werden und auf Basis des bisherigen Studiums nicht unbedingt jedem Teilnehmer bekannt sind.

2.1 Hashfunktionen und universelle Hashfamilien

Da Hashfunktionen aus universellen Hashfamilien verwendet werden um zu beweisen, dass die amortisierte erwartete Einfügezeit konstant ist, erläutere ich hier kurz was eine solche Familie auszeichnet.

Die Kenntnis über universelle Hashfamilien ist für den Beweis der Laufzeit zum Einfügen wichtig. Auf Grund dessen werde ich diese in meinem Vortrag vorstellen.

2.2 Kuckucksgraph

Hier führe ich eine Definition des Kuckucksgraphen ein um diesen später bei der Beschreibung des Kuckuckshashing verwenden zu können. Damit lassen sich die verschiedenen Fälle beim Einfügen eines neuen Schlüssels gut erläutern.

Kuckucksgraphen sind zum Verständnis des Themas nicht wichtig. Daher werde ich diesen in meinem Vortrag eine niedrige Priorität geben.

3 Kuckuckshashing

Das ist der wichtigste Abschnitt, da hier das eigentliche Thema der Ausarbeitung tiefergehend behandelt wird.

3.1 Funktionsweise

Hier erkläre ich die Funktionsweise des normalen, symmetrischen Kuckuckshashing ein. Ich erkläre wie die Operationen Lookup, Insert und Delete dabei umgesetzt werden. Verbunden mit der Lookup Operation werde ich zur Visualisierung Beispiele zeigen. Ich werde auch den Bezug zum Rehashing aufbauen. Zu jeder der Operationen werde ich eine Implementierung im Pseudocode zeigen.

Dieser Unterabschnitt wird für den Vortrag am höchsten priorisiert, da hier das Kuckuckshashing wirklich erklärt wird.

3.2 Laufzeitanalyse

In diesem Unterabschnitt werde ich eine Analyse der Laufzeit vornehmen. Dabei werde ich kurz auf die Laufzeit der Operationen Lookup und Delete eingehen und dann zu Insert kommen. Hierfür werde ich dann einen Beweis der konstanten erwarteten amortisierten Zeit nachvollziehen, die zum Einfügen eines Elements benötigt wird. Dazu werde ich auch das Rehashing betrachten. Abschließend gehe ich kurz auf die De-Amortisierung des Kuckuckshashing unter Verwendung einer Queue ein.

Auf die Laufzeitanalyse werde ich im Vortrag auch eingehen, aber sie nicht so tief behandeln, wie in der Ausarbeitung.

3.3 Variationen und Anwendungen

In diesem Unterabschnitt gehe ich auf eine Auswahl verschiedener Variationen von Kuckuckshashing ein. Dabei werde ich auf asymmetrisches Kuckuckshashing, die Erweiterung um zusätzliche Tabellen und Kuckuckshashing mit einem Stapel eingehen. Ich werde hier eventuelle Vor- und Nachteile betrachten. Außerdem spreche ich den Kuckucksfilter an, welcher eine Erweiterung des Bloom Filters darstellt und vom Konzept des Kuckuckshashing Gebrauch macht.

Auf die Inhalte dieses Abschnitts werde ich in meinem Vortrag kurz eingehen, aber nicht allzu ausführlich.

4 Vergleich mit anderen Hashmethoden

Hier werde ich andere Hashmethoden für einen Vergleich der Laufzeit der Operationen heranziehen und zeigen wie Kuckuckshashing im Vergleich zu anderen

Methoden abschneidet. Für diesen Vergleich werde ich die populären Hashmethoden "Hashing mit Verkettung", "Doppel-Hashing" vorstellen.

Um abschätzen zu können, wie sinnvoll der Einsatz von Kuckuckshashing ist, ist es sinnvoll einen Vergleich zu anderen Hashmethoden ziehen zu können, daher werde ich die Erkenntnisse aus diesem Abschnitt für den Vortrag höher priorisieren.

5 Fazit

Dieser Abschnitt enthält eine abschließende Zusammenfassung der Ergebnisse. Zusätzlich möchte ich hier auch kurz auf die Praxistauglichkeit von Kuckuckshashing eingehen.