Hochschule Aalen
Fakultät Elektronik und Informatik
Studienbereich Informatik

Algorithmen und Datenstrukturen 2

Vorlesung im Sommersemester 2024
Prof. Dr. habil. Christian Heinlein

1. Praktikumsaufgabe (11. April – 2. Mai 2024)

Aufgabe 1: Streuwerttabellen

Auf der Vorlesungswebseite befindet sich eine Datei `hashing.h` mit folgenden unvollständigen Klassen (bzw. Strukturen), die jeweils als Schablonen (templates) mit geeigneten Typparametern definiert sind:

- `HashChain<K, V>` implementiert Streuwerttabellen mit Schlüsseltyp `K` und Werttyp `V` mittels Verkettung.
  Hierfür muss eine Funktion `hashval` mit Parametertyp `K` und Resultattyp `uint` (`unsigned int`) zur Berechnung von Streuwerten sowie ein Gleichheitsoperator für den Typ `K` verfügbar sein.

- `LinProb<K>`, `QuadProb<K>` und `DblHash<K>` implementieren Sondierungssequenzen mit Schlüsseltyp `K` für lineare bzw. quadratische Sondierung bzw. doppelte Streuung.
  Hierfür muss ebenfalls eine derartige Funktion `hashval` verfügbar sein. Für `DblHash` muss außerdem eine zweite Streuwertfunktion `hashval2` mit Parametertypen `K` und `uint` (der zweite Parameter gibt die Tabellengröße an) und Resultattyp `uint` verfügbar sein.

- `HashOpen<K, V, S>` implementiert Streuwerttabellen mit Schlüsseltyp `K` und Werttyp `V` mittels offener Adressierung mit Sondierungssequenzen des Typs `S` (für den also insbesondere `LinProb<K>`, `QuadProb<K>` und `DblHash<K>` eingesetzt werden können).

Implementieren Sie alle leeren und fehlenden Konstruktoren und Elementfunktionen dieser Klassen sowie die hierfür erforderlichen Datenstrukturen, das heißt:

- Den Konstruktor und die Elementfunktionen `put`, `get`, `remove` und `dump` der Klassen `HashChain` und `HashOpen`.

- Den Konstruktor und die Elementfunktion `next` der Klassen `LinProb`, `QuadProb` und `DblHash`.

Beachten Sie unbedingt die in der Datei enthaltenen Kommentare, die das zu implementierende Verhalten spezifizieren! Beachten Sie auch die Hinweise zur dynamischen Erzeugung von Feldern in §1.4.8 der Vorlesungsfolien!

Um automatisierte Tests der Implementierungen zu ermöglichen, dürfen die vorgegebenen Klassennamen und die Signaturen der Elementfunktionen nicht verändert werden, und es dürfen keine Diagnoseausgaben produziert werden.

Abgesehen von `<iostream>`, dürfen keine Bestandteile der C++-Standardbibliothek oder anderer Bibliotheken verwendet werden.

Abzugeben ist die entsprechend erweiterte Datei `hashing.h`.
Die E-Mail mit der Abgabe muss als Betreff `Algo2 Gruppe NN` mit zweistelliger Gruppennummer `NN` (z. B. `05` oder `12`) und als Anhang die abzugebende Datei haben. Der Nachrichtentext muss für jedes Gruppenmitglied aus einer Zeile der Art `Vorname,Nachname,Matrikelnummer` bestehen. (Auch wenn eine Lösung mehrmals abgegeben wird, muss die E-Mail jedesmal genau dieses Format haben. Es wird dann automatisch die letzte fristgerechte Abgabe der Gruppe gewertet.)

Testprogramm

Die ebenfalls auf der Vorlesungswebseite verfügbare Datei `hashtest.cxx` enthält ein einfaches interaktives Testprogramm.

Das Shellskript `hashtest.sh` ruft dieses Testprogramm mit einigen Befehlen jeweils für Verkettung, lineare Sondierung, quadratische Sondierung und doppelte Streuung auf. Bei korrekter Implementierung aller Klassen muss der Befehl `sh hashtest.sh` in einer Linux-Konsole exakt die folgende Ausgabe liefern:

```
c
4 (1, 3) eins
6 (3, 3) drei
6 (2, 4) zwei
l
4
5 (1, 3) eins
6 (2, 4) zwei
7 (3, 3) drei
q
4
5 (1, 3) eins
6 (2, 4) zwei
7 (3, 3) drei
d
1 (1, 3) eins
4
5 (3, 3) drei
6 (2, 4) zwei
```

Organisatorische Hinweise

Beachten Sie hierzu das Dokument „Wichtige Hinweise zur Prüfung“ auf der Vorlesungswebseite!