

Hinweise:

- Die Übungsblätter sollen in Gruppen von je 3 Studierenden aus der gleichen Kleingruppenübung bearbeitet werden.
- Die Lösungen müssen bis Montag, den 21. Juni um 11:00 Uhr in den entsprechenden Übungskästen einge-worfen werden. Sie finden die Kästen am Eingang Halifaxstr. des Informatikzentrums (Ahornstr. 55).
- Namen und Matrikelnummern der Studierenden sowie die Nummer der Übungsgruppe sind auf jedes Blatt der Abgabe zu schreiben. Heften bzw. tackern Sie die Blätter!
- An den Freitagen **18. und 25. Juni** findet, **keine** Vorlesung statt.
- Die Globalübung am 14. Juni findet ausnahmsweise im Raum H218 (Intzestrasse 5) statt.

Aufgabe 1 (Hashing): **(4 + 12 + 2 + 2 + 2 Punkte)**

In einem kleinen Studentenkino mit $m = 23$ Plätzen wird ein Film gezeigt. Um dem Ansturm Herr zu werden, sollen die Plätze mit einem offenen Hashverfahren auf die Wartenden verteilt werden. Als Schlüssel werden dabei nur die beiden letzten Ziffern der Matrikelnummer verwendet. Betrachten Sie die folgenden beiden Hashfunktionen:

- $h_1(x) :=$ Quersumme von x
- $h_2(x) := x \bmod 23$ (Division-Rest-Methode)

- a)** Diskutieren Sie, inwieweit h_1 und h_2 die Bedingungen, die an eine sinnvolle Hashfunktion gestellt werden, erfüllen.
- b)** Welche Platznummern erhalten die Besucher, wenn sie in der gegebenen Reihenfolge das Kino betreten?

6, 16, 61, 87, 69, 90, 4, 43, 57, 4, 12, 80, 46

Verwenden Sie jeweils folgende Hashfunktionen:

- lineares Sondieren mit h_1 und mit h_2 als Hashfunktion
 - quadratisches Sondieren mit h_1 und mit h_2 als Hashfunktion und mit $c_1 = 2$ und $c_2 = 3$ als Konstanten,
 - Doppelhashing mit h_1 als erster und h_2 als zweiter Hashfunktion. Inwieweit muss h_2 geändert werden, um Probleme zu vermeiden?
- c)** Warum ist ein geschlossenes Hashing, in dem geschildertem Szenario, nicht praktikabel bzw. sinnvoll?
- d)** Angenommen wir haben eine perfekte Hashfunktion und fügen die obigen Elemente in einen Hash der Größe 23 ein. Wieviele Sondierungen sind dann durchschnittlich bei erfolgreicher Suche nötig?
- e)** Angenommen wir haben eine perfekte Hashfunktion und fügen die obigen Elemente in einen Hash der Größe 23 ein. Wieviele Sondierungen sind dann durchschnittlich bei nicht erfolgreicher Suche nötig?

Aufgabe 2 (gutes Hashing):

(5 Punkte)

Diskutieren Sie, welche Eigenschaften eine gute Hashfunktion für Strings haben sollte. Es sei m die Größe der Hashtabelle und $s = a_1 a_2 \dots a_n$ ein String, wobei wir die Zeichen mit ihren ASCII-Codes identifizieren. Wie gut sind die folgenden drei Hashfunktionen:

$$\left(\sum_{k=1}^n a_k \right) \bmod m \quad \left(\sum_{k=1}^n k \cdot a_k \right) \bmod m \quad a_1^{a_2^{a_3}} \bmod m$$

Aufgabe 3 (Countingsort):

(6 + 2 Punkte)

- a) Das folgendes Array ist mit Countingsort zu sortieren. Geben Sie das Histogramm- und das Positionsarray vor dem ersten Einfügen ins Ausgabearray an, sowie das Positions- und Ausgabearray nach jedem Einfügeschritt.

4	3	0	1	4	2	3	7	3
---	---	---	---	---	---	---	---	---

- b) Der in der Vorlesung vorgestellte Algorithmus Countingsort fügt die Elemente des Eingabearrays von hinten nach vorne in das Ausgabearray ein. Welche Nachteile ergäben sich, wenn man das Eingabearray stattdessen von vorne nach hinten durchlaufen würde?