

Institut für Theoretische Informatik  
Peter Widmayer  
Sandro Montanari  
Tobias Pröger

13. März 2013

## Datenstrukturen & Algorithmen      Programmieraufgabe 4      FS 13

In dieser Aufgabe soll offenes Hashing programmiert werden. Konkret sind  $n$  Schlüssel  $k_1, \dots, k_n$  gegeben, die in ein initial leeres Array der Grösse  $m \geq n$  eingefügt werden sollen, wobei  $m$  eine Primzahl ist. Das Element  $k_i$  wird in der Position

$$h(k_i) = k_i \bmod m$$

gespeichert, falls diese frei ist. Ansonsten gibt es eine Kollision und  $k_i$  muss in einer anderen Position des Arrays gespeichert werden. Nach  $j$  Kollisionen betrachten wir die Position

$$h(k_i) - s(j, k_i)$$

und fügen  $k_i$  dort ein, falls sie leer ist. Wir betrachten die folgenden Sondierungsfunktionen, die auch in der Vorlesung vorgestellt wurden:

- $s(j, k_i) = j$  (Lineares Sondieren),
- $s(j, k_i) = \lceil \frac{j}{2} \rceil^2 \cdot (-1)^j$  (Quadratisches Sondieren),
- $s(j, k_i) = j \cdot h'(k_i)$  mit  $h'(k_i) = 1 + (k_i \bmod (m - 2))$  (Double Hashing).

**Eingabe** Die erste Zeile der Eingabe enthält lediglich die Anzahl  $t$  der Testinstanzen. Es folgt eine Zeile für jede Testinstanz mit den Zahlen  $m, n, k_1, \dots, k_n$  mit einer Primzahl  $m \geq n$ . Die Elemente  $k_1, \dots, k_n$  werden in dieser Reihenfolge in je drei initial leere Hashtabellen eingefügt, und zwar je eine für jede der oben genannten Sondierungsfunktionen.

**Ausgabe** Für jede Testinstanz geben wir genau drei Zeilen aus. Die erste Zeile startet mit der Gesamtanzahl von Kollisionen, wenn *lineares Sondieren* verwendet wird, gefolgt von dem Inhalt der Hashtabelle nachdem alle Schlüssel  $k_1, \dots, k_n$  eingefügt wurden. Eine leere Position in der Tabelle soll durch den Wert 0 signalisiert werden. Die zweite und die dritte Zeile sind analog aufgebaut und enthalten die Ergebnisse, wenn quadratisches Sondieren (zweite Zeile) bzw. Double Hashing (dritte Zeile) benutzt werden.

### Beispiel

*Eingabe:*

---

```
2
5 3 1 2 3
5 4 1 2 3 8
```

---

*Ausgabe:*

---

```
0 0 1 2 3 0
0 0 1 2 3 0
0 0 1 2 3 0
3 8 1 2 3 0
1 0 1 2 3 8
1 8 1 2 3 0
```

---

**Abgabe:** Bis Mittwoch, den 20. März 2013.