

---

## Grundlagen: Algorithmen und Datenstrukturen

---

*Abgabetermin: 26. August 2011*

### Hausaufgabe 1

In der Vorlesung wurde vorgestellt, wie man ein dynamisches Array implementieren kann, so dass jede Operation amortisiert höchstens  $\mathcal{O}(1)$  Zeit verbraucht. Implementieren Sie die auf der Übungswebseite bereitgestellte abstrakte Klasse `UArray` in Java in der Klasse `UIArray`. Bitte verwenden Sie in den Hausaufgaben, wie in der Vorlesung, für `alpha` bzw. `beta` die Werte 4 bzw. 2.

### Hausaufgabe 2

Erweitern Sie Ihre Implementierung aus Hausaufgabe 1 so zu einer Klasse `UIcArray`, dass die Operationen Ihrer Implementierung nicht nur amortisiert, sondern auch im Worst-Case, höchstens  $\mathcal{O}(1)$  Zeit benötigen.

*Hinweis:* Sie dürfen zusätzliche Arrays benutzen. Kopieren Sie geschickter als in der ersten Implementierung.

### Hausaufgabe 3

Implementieren Sie eine Warteschlange auf der Basis eines Dynamischen Arrays.

Erweitern Sie die abstrakte Klasse `UQueue` zu einer Klasse `UIQueue`.

Verwenden Sie die auf der Übungsseite bereitgestellte Klasse `UIsArray` und modifizieren Sie diese *nicht*.

### Aufgabe 1

In der Vorlesung wurde angesprochen, dass ein Dynamisches Array erst verkleinert werden darf, wenn das Array nur noch zur zu einem Viertel gefüllt ist. Beweisen Sie:

Wird das Array schon verkleinert, wenn der Füllstand des Arrays nur noch die Hälfte des reservierten Speicherplatzes beträgt, so stimmt die Behauptung nicht, dass jede Folge von  $m$  Operationen auf dem Array in Zeit  $\mathcal{O}(m)$  abgearbeitet werden kann.

### Aufgabe 2

Bestimmen Sie die Wahrscheinlichkeit, dass von  $n$  Personen mindestens zwei Personen am gleichen Tag im Jahr Geburtstag haben.

Erläutern Sie den Zusammenhang zwischen Hashing und dem eben gezeigten Geburtstagsparadoxon.

*Hinweis:* Bestimmen Sie die Wahrscheinlichkeit, des Komplementär-Ereignisses: Keine zwei Person haben am gleichen Tag Geburtstag. Ein Jahr hat hier immer 365 Tage.

### Aufgabe 3

Veranschaulichen Sie Hashing mit Chaining und Hashing mit Linear Probing. Fügen Sie dazu die Elemente

$$3, 11, 9, 7, 14, 23, 4, 12, 8, 15, 0$$

in eine Hash-Tabelle der Größe 11 ein.

Benutzen Sie die Hashfunktion

$$h_5(x) = 5x \bmod 11.$$

### Aufgabe 4 (10 Punkte)

Die Werte {Apfel, Banane, Kirsche, Himbeere, Melone} sollen in einer Hashtabelle der Größe  $n = 4$  untergebracht werden. Es seien folgende Hashfunktionen gegeben:

$f_1$ : Apfel $\mapsto$ 4	Banane $\mapsto$ 2	Kirsche $\mapsto$ 2	Himbeere $\mapsto$ 1	Melone $\mapsto$ 4
$f_2$ : Apfel $\mapsto$ 3	Banane $\mapsto$ 4	Kirsche $\mapsto$ 2	Himbeere $\mapsto$ 3	Melone $\mapsto$ 4
$f_3$ : Apfel $\mapsto$ 2	Banane $\mapsto$ 2	Kirsche $\mapsto$ 4	Himbeere $\mapsto$ 1	Melone $\mapsto$ 1
$f_4$ : Apfel $\mapsto$ 1	Banane $\mapsto$ 3	Kirsche $\mapsto$ 3	Himbeere $\mapsto$ 4	Melone $\mapsto$ 4
$g_1$ : Apfel $\mapsto$ 1	Banane $\mapsto$ 1	Kirsche $\mapsto$ 3	Himbeere $\mapsto$ 2	Melone $\mapsto$ 3
$g_2$ : Apfel $\mapsto$ 2	Banane $\mapsto$ 4	Kirsche $\mapsto$ 2	Himbeere $\mapsto$ 3	Melone $\mapsto$ 4
$g_3$ : Apfel $\mapsto$ 4	Banane $\mapsto$ 4	Kirsche $\mapsto$ 1	Himbeere $\mapsto$ 4	Melone $\mapsto$ 2
$g_4$ : Apfel $\mapsto$ 3	Banane $\mapsto$ 1	Kirsche $\mapsto$ 2	Himbeere $\mapsto$ 3	Melone $\mapsto$ 3
$g_5$ : Apfel $\mapsto$ 4	Banane $\mapsto$ 2	Kirsche $\mapsto$ 2	Himbeere $\mapsto$ 2	Melone $\mapsto$ 3

In der Vorlesung haben wir den Begriff der  $c$ -universellen Hashfunktionen kennengelernt.

- Geben Sie für die Familie  $\mathcal{H}_1 = \{f_1, f_2, f_3, f_4\}$  das kleinste  $c$  an, so dass  $\mathcal{H}_1$   $c$ -universell ist.
- Finden Sie eine möglichst kleine Familie  $\mathcal{H}_2 \subseteq \{g_1, g_2, g_3, g_4, g_5\}$ , die 1-universell ist.

Begründen Sie Ihre Aussagen.