Ausgabedatum: 13.12.2012
Version: 2012-12-13 16:21

# 10. Übungsblatt zur Vorlesung Entwurf und Analyse von Algorithmen, WS 12/13

***Abgabe:*** *Bis Donnerstag, 20.12.2012, 12:00 Uhr, Abgabenkasten im Treppenhaus 48-6.*

## 59. Aufgabe

**Track $\varepsilon$** 4
**Track AI** [4]

Wir betrachten Hashing mit Indirect Chaining und nehmen zur Vereinfachung an, $\mathsf{max} = m + 1$ sei eine gerade Zahl. Entgegen der bisherigen Annahmen seien nicht alle Hash-Folgen $a_1, a_2, \ldots, a_n$ gleichwahrscheinlich, sondern eine gerade Adresse (und die Null) werde doppelt so häufig durch unsere Hash-Funktion adressiert wie eine ungerade. Alle geraden Adressen (und die Null) untereinander seien gleichwahrscheinlich, ebenso alle ungeraden.

Bestimmen Sie unter dieser Annahme die erwarteten Kosten für eine erfolglose Suche in einer Hash-Tabelle mit $n$ Schlüsseln. Dabei werde auch für die Suche eine gerade Adresse (und die Null) doppelt so häufig aufgesucht wie eine ungerade.

**Hinweis:** Zur Lösung dieser Aufgabe bietet es sich an, zuerst die Analyse für die uniforme Verteilung aller Hash-Folgen zu erarbeiten, um dann Veränderungen an den passenden Stellen vorzunehmen.

## 60. Aufgabe

**Track $\varepsilon$** [3]
**Track AI** [3]

Wir betrachten Hashing mit Indirect Chaining und ersetzen die linearen Listen durch binäre Suchbäume. Wie in der Vorlesung sei $\alpha = \frac{N}{\max}$ und wir nehmen an, dass alle $\max^N$ Hashfolgen gleichwahrscheinlich sind. Zeigen Sie, dass für die Zahl an Vergleichen bei der erfolglosen Suche $\mathrm{CU}_N$ gilt, dass

$$\mathbb{E}[\mathrm{CU}_N] \sim 2\log(\alpha) ,$$

wobei $\alpha \to \infty$.

**Hinweis:**

(i) Für $P_{N,k}$ die Wahrscheinlichkeit, dass ein Baum genau $k$ der $N$ Schlüssel enthält, gilt:

$$\sum_{k=0}^{\infty}\sum_{i=1}^{k}\frac{P_{N,k}}{i} = \log(\alpha) + \mathcal{O}(1)$$

für $\alpha \to \infty$.

(ii) Aus der Gleichwahrscheinlichkeit der Hashfolgen ergibt sich, dass alle Endpositionen der erfolglosen Suche in jedem (festen) Kollisionsbaum gleich wahrscheinlich sind[1].

## 61. Aufgabe

| | |
|---|---|
| **Track $\varepsilon$** | [2] |
| **Track AI** | [2] |

Zeigen Sie, dass für eine geeignete Konstante $c$ und für alle $N, m \in \mathbb{N}$ die Menge aller Hash-Funktionen $h : [1 \dots N] \to [0 \dots m]$ $c$-universell ist. Bestimmen Sie den minimalen Wert von $c$.

## 62. Aufgabe

| | |
|---|---|
| **Track $\varepsilon$** | 2 |
| **Track AI** | 2 |

Gegeben sei ein markierter Digraph $G = (V, E, r)$ mit $r : E \to [0, 1] \subseteq \mathbb{R}$. Wir interpretieren die Kanten $e \in E$ als Leitungen eines Kommunikationsnetzwerks und deren Markierung als die *Zuverlässigkeit* der entsprechenden Verbindung (Kante), indem wir $r(e)$ als die Wahrscheinlichkeit dafür auffassen, dass eine Kommunikation über Verbindung $e$ nicht scheitert. Dabei nehmen wir an, dass alle Wahrscheinlichkeiten unabhängig voneinander sind.

Entwerfen Sie einen Algorithmus, der in Zeit $\mathcal{O}(|V|^2)$ die verlässlichste Verbindung zwischen zwei gegebenen Knoten berechnet.

## 63. Aufgabe

| | **a)** | **b)** |
|---|---|---|
| **Track $\varepsilon$** | 1 | 3 |
| **Track AI** | 1 | 3 |

Ein *Superstar* ist eine Person, die von allen anderen Personen gekannt wird, die selbst aber keine andere Person kennt.

a) Wenn Sie eine Menge von $n$ Personen und die Relation „$a$ kennt $b$“ als Digraph modellieren, also „$a$ kennt $b$“ $\iff$ $(a, b) \in E$, wie können Sie dann einen Superstar charakterisieren?

[1] Können Sie das zeigen?

b) Sei der Digraph aus Teilaufgabe a) als $n \times n$ Adjazenzmatrix $A$ gegeben. Entwerfen Sie einen Algorithmus, der bei Eingabe $A$ in Zeit $o(n^2)$ entscheidet, ob die durch $A$ modellierte Population einen Superstar besitzt.

Begründen Sie die Korrektheit Ihres Algorithmus und weisen Sie eine dem Zwecke der Aufgabenstellung dienliche Laufzeitschranke nach.

## 64. Aufgabe

| | a) | b) |
|---|---|---|
| **Track $\varepsilon$** | [2] | [3] |
| **Track AI** | 2 | [3] |

a) Für einen Digraphen $G = (V, E)$ ist seine *transitive Hülle* der Digraph $G' = (V, E')$ mit $e = (u, v) \in E'$ genau dann, wenn es in $G$ einen gerichteten Weg von $u$ nach $v$ gibt.

Sei ein Digraph $G = (V, E)$ gegeben und $A$ dessen Adjazenzmatrix. Wie können Sie durch Potenzierung der (ggf. modifizierten) Matrix $A$ die transitive Hülle von $G$ berechnen?

b) Sei ein Digraph $G = (V, E)$ mit $V = \{1, 2, \ldots, n\}$ gegeben und `A` dessen Adjazenzmatrix. Beweisen Sie, dass folgender Algorithmus (Warshalls-Algorithmus)

```
FOR i:=1 TO n DO
  FOR s:= 1 TO n DO
      FOR t:= 1 TO n DO
          IF (A[s,i]=1) AND (A[i,t]=1) THEN A[s,t]:= 1; (*)
      END;
  END;
END;
```

die transitive Hülle von $G$ berechnet. (Damit können wir mit einer Preprocessing-Zeit in $\mathcal{O}(n^3)$ eine Datenstruktur zur Verfügung stellen, die den Test auf Erreichbarkeit in Digraphen in konstanter Zeit unterstützt.)

## 65. Aufgabe

| | |
|---|---|
| **Track $\varepsilon$** | [1] |
| **Track AI** | 1 |

Zur Erinnerung: Der Durchmesser eines (Di)Graphen $G = (V, E)$ ist definiert als

$$d(G) := \max \{\mathrm{dist}(u, v) \mid u, v \in V\} \ ,$$

wobei $\mathrm{dist}(u, v)$ in unmarkierten (Di)Graphen die Anzahl Kanten auf einem kürzesten Weg von $u$ nach $v$ ist.

Entwerfen Sie einen Algorithmus, der in Zeit $\Theta(|V|^3)$ den Durchmesser eines beliebigen unmarkierten (Di)Graphen findet. Beweisen Sie die Korrektheit und Laufzeitschätzung Ihres Entwurfs.