Prof. Dr. Matthias Rarey
M. v. Behren, F. Heitmann, M. Hilbig,
F. Lauck, T. Otto, J. Röwekamp

Zentrum für Bioinformatik Hamburg (ZBH)
Abteilung für Algorithmisches Molekulares Design

Wintersemester 2012/2013

Übung zur Vorlesung Algorithmen und Datenstrukturen
Übung 5

Abgabe: 17.12.2012, 12 Uhr

Aufgabe 1: Hashing

a)

Fügen Sie die folgenden Werte in eine Hashtabelle der Größe 13 ein: $18, 4, 30, 8, 64, 43$. Verwenden Sie die Hash-Funktionen $h_1(k) = k \bmod 13$ und $h_2(k) = 1 + (k \bmod 5)$ zusammen mit den folgenden Verfahren:

- Verkettung mit $h_1(k)$.
- Offene Adressierung mit linearer Sondierung und der Hash-Funktion:
  $h(k,i) = (h_1(k) + i) \bmod 13$
- Offene Adressierung mit doppeltem Hashing und der Hash-Funktion:
  $h(k,i) = (h_1(k) + ih_2(k)) \bmod 13$

Geben Sie die Hash-Tabellen jeweils am Ende aller Einfügeoperationen an.
Geben Sie bei den Aufgabenteilen, welche Sondierung erfordern, ebenfalls die Sondierungsreihenfolge an! (2 Punkte)

b)

Gegeben seien die folgenden drei Hash-Funktionen, die die nötige Auswertung einer zusätzlichen Modulo-Operation vermeiden:

- $h_1(k) = k \bmod m$
- $h_2(k) = 1 + b \cdot h_1(k)$
- $h(k,i) = (h_1(k) + ih_2(k)) \bmod m$

Erläutern Sie kurz, ob $h(k,i)$ sinnvoll als Hash-Funktion mit doppeltem Hashing verwendet werden kann. Begründen Sie Ihre Antwort! *Hinweis:* Überlegen Sie sich zunächst warum das doppelte Hashing überhaupt betrachtet wurde. Danach überlegen Sie sich, ob dies durch die gegebenen Funktionen erfüllt werden kann.

Ändert sich die Antwort, wenn $h_2(k)$ wie folgt abgewandelt wird? (Begründung!)

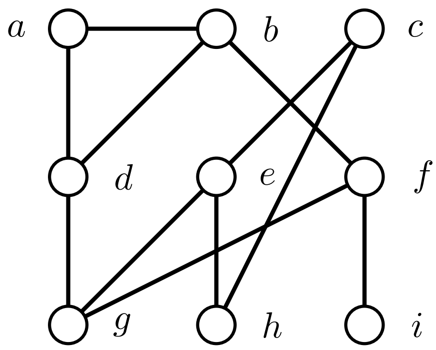
- $h_2(k) = 1 + k \cdot h_1(k)$

(3 Punkte)

Aufgabe 2: Breitensuche

a)

Berechnen Sie mit Hilfe der Breitensuche die Länge des kürzesten Weges von Knoten d zu Knoten i.
Geben Sie den Graphen nach jeder Iteration inklusive der korrekten Färbung und den aktuellen Werten von Q an!



(2 Punkte)

b)

Ein ungerichteter, ungewichteter Graph $G = (V, E)$ (V bezeichne die Knotenmenge und E die Kantenmenge) wird als *verbundener Doppelgraph* bezeichnet, wenn er (gleichzeitig) die folgenden Bedingungen erfüllt:

- $V = V_1 \cup V_2$
- $u, v \in V_1 \Rightarrow (u, v) \notin E$
- $w, x \in V_2 \Rightarrow (w, x) \notin E$
- $V_1 \cap V_2 = \emptyset$

Entwerfen Sie auf Basis der Breitensuche einen Algorithmus, der überprüft, ob es sich bei einem gegebenen ungerichteten, ungewichteten Graphen um einen *verbundenen Doppelgraphen* handelt. Ihr Algorithmus soll in $O(M + N)$ Zeitschritten zu einem Ergebnis kommen. (Dabei bezeichne M wie üblich die Kantenzahl und N die Knotenzahl des Graphen)
Erläutern Sie (zusätzlich zum Pseudocode) die grundlegende Idee hinter Ihrem Algorithmus und begründen Sie informell aber schlüssig, warum er korrekt ist und die geforderte Laufzeit hat.

(3 Punkte)

Aufgabe 3: Tiefensuche

a)

Gegeben sei der folgende gerichtete, ungewichtete Graph $G = (V, E)$ in seiner formalen Notation:

- $V = \{a, b, c, d\}$
- $E = \{(b, b), (a, c), (b, c), (c, d), (d, b), (c, b)\}$

Stellen Sie den Graphen als Zeichnung, in Form einer Adjazenzmatrix und in Form von Adjazenzlisten dar. (1 Punkt)

b)

Führen Sie auf dem Graphen aus Aufgabe 3a) den Tiefensuche-Algorithmus aus und geben Sie die graphische Darstellung des Graphen nach jedem Iterationsschritt an! Sie können einen beliebigen Knoten als Startknoten verwenden. Gehen Sie davon aus, dass die Knoten und die Adjazenzlisten lexikographisch sortiert sind.

(2 Punkte)

c)

Entwickeln Sie einen Algorithmus auf Basis der Tiefensuche, der für einen gegebenen gerichteten Graphen in $O(N + M)$ Zeit entscheidet, ob der Graph einen Zyklus enthält. Falls ja, soll ihr Algorithmus einen Zyklus als Knotenliste ausgeben und terminieren. Skizzieren Sie Ihre Lösungsidee und beschreiben Sie Ihren Algorithmus in kommentiertem Pseudocode.

(3 Punkte)