

## 1. Übungsblatt

**Ausgabe:** 19. April 2005

**Abgabe:** 26. April in der Vorlesung

Die Bearbeitung in Zweiergruppen ist ausdrücklich erwünscht.

### Problem 1: Approximationsalgorithmen

\*\*\*

---

#### Algorithmus 1 : Approximationsalgorithmus für VERTEX COVER

---

**Eingabe :** Graph  $G = (V, E)$

**Ausgabe :**  $S$  (Vertex Cover von  $G$ )

$S := \emptyset$

**Für** alle Kanten  $\{v_i, w_i\} \in E$

**Wenn**  $\{v_i, w_i\}$  nicht durch  $S$  überdeckt ist  
    |  $S \leftarrow v_i$   
    |  $S \leftarrow w_i$

---

Zeigen Sie, dass Algorithmus 1 ein Faktor-2-Approximationsalgorithmus ist.

### Problem 2: Aufspannende Bäume I

\*\*

Zeigen Sie, dass die folgenden Aussagen für einen ungerichteten Graphen  $G = (V, E)$  äquivalent sind ( $n := |V|, m := |E|$ ).

- (a)  $G$  ist ein Baum, d.h.  $G$  ist zusammenhängend und enthält keine Kreise.
- (b) Zwischen je zwei Knoten in  $V$  gibt es genau einen Weg in  $G$ .
- (c)  $G$  ist zusammenhängend und  $m = n - 1$ .

### Problem 3: Aufspannende Bäume II

\*

Sei  $G = (V, E)$  ein Baum,  $u, v \in V$ ,  $v_0, v_1, \dots, v_k$  ein Pfad in  $G$ . Dann gilt: Der Graph  $G' = (V, E')$  mit  $E' = E \setminus \{v_i, v_{i+1}\} \cup \{v_0, v_k\}$  ist ein Baum.

## Problem 4: Verteilte Graphenalgorithmen

\*\*\*

Finden Sie heraus, was Algorithmus 2 tut und bestimmen Sie seine Laufzeit im synchronen Modell. Der Algorithmus wird gestartet, indem die Nachricht  $\langle M, \text{start} \rangle$  an einen Knoten geschickt wird.

---

### Algorithmus 2 : Code für Prozessor $p$

---

**Initialisierung** :  $\text{Vor} := \text{null}$ ,  $\text{Nach} := \emptyset$ ,  $\text{Quer} := \emptyset$

**Beim Empfang von**  $\langle M, \text{start} \rangle$

└ Sende  $\langle M \rangle$  an alle Nachbarn  
└  $\text{Vor} := p$

**Beim Empfang von**  $\langle M \rangle$  von Nachbar  $q$

└ **Wenn**  $\text{Vor} == \text{null}$

└└  $\text{Vor} := q$

└└ sende  $\langle M \rangle$  an alle Nachbarn außer  $q$

└└ **Wenn**  $\text{Nachbarn}(p) == \{q\}$

└└└ Fertig()

└ sonst sende  $\langle \text{bekannt} \rangle$  an  $q$

**Beim Empfang von**  $\langle \text{fertig} \rangle$  von  $q$

└  $\text{Nach} \leftarrow q$

└ **Wenn**  $\text{Nach} \cup \text{Quer} == \text{Nachbarn}(p) \setminus \text{Vor}$

└└ Fertig()

**Beim Empfang von**  $\langle \text{bekannt} \rangle$  von  $q$

└  $\text{Quer} \leftarrow q$

└ **Wenn**  $\text{Nach} \cup \text{Quer} == \text{Nachbarn}(p) \setminus \text{Vor}$

└└ Fertig()

**Funktion**  $\text{Fertig}()$

└ sende  $\langle \text{fertig} \rangle$  an  $\text{Vor}$

└ terminiere

---