

6. Übungsblatt

Ausgabe: 30. Januar 2007

Abgabe: 6. Februar, 15:30 Uhr im ITI Wagner (Informatik-Hauptgebäude, 3. Stock)

Die Bearbeitung in Zweiergruppen ist ausdrücklich erwünscht.

Problem 1: Fingerabdrücke

2pt

Mit Hilfe von *Fingerabdrücken* kann man die Gleichheit von Objekten prüfen, indem man statt der gesamten Struktur der Objekte nur einen Teil, den sogenannten Fingerabdruck, vergleicht. Sind die Fingerabdrücke der Objekte verschieden so weiß man, dass auch die Objekte verschieden sind, der Umkehrschluss gilt im Allgemeinen nicht. Gegeben sei Algorithmus 1, der überprüft, ob für drei $n \times n$ reelle Matrizen A, B, C gilt: $AB = C$.

Algorithmus 1 : MATRIZENPRODUKTTEST (A, B, C)

```
1  $r \leftarrow \langle$  Vektor von  $n$  unabhängigen Zufallsbits  $\rangle$ 
2  $x \leftarrow Br$ 
3  $y \leftarrow Ax$ 
4  $z \leftarrow Cr$ 
5 Wenn  $y \neq z$ 
6   | return NO
7 sonst
8   | return YES
```

- Zeigen Sie, dass Algorithmus 1 YES liefert, wenn gilt $AB = C$.
- Zeigen Sie, dass die Wahrscheinlichkeit, dass Algorithmus 1 YES liefert, obwohl $AB \neq C$ gilt, kleiner oder gleich $1/2$ ist.
- Wie kann diese Wahrscheinlichkeit einer fälschlichen Ausgabe YES (siehe (b)) einfach reduziert werden?
(*Hinweis*: ohne Beweis)

Problem 2: Approximation der Größe einer Clique

3pt

Eine Clique ist ein vollständig verbundener Subgraph eines Graphen. Sei $G = (V, E)$ ein ungerichteter, zusammenhängender, einfacher Graph. Für jedes $k \geq 1$, $k \in \mathbb{N}$ sei $G^{(k)}$ der ungerichtete Graph $(V^{(k)}, E^{(k)})$, wobei $V^{(k)}$ die Menge aller geordneten k -Tupel von Knoten aus V ist und $E^{(k)}$ so definiert ist, dass (v_1, v_2, \dots, v_k) genau dann mit (w_1, w_2, \dots, w_k) benachbart ist, wenn für alle i mit $1 \leq i \leq k$ entweder der Knoten v_i in G mit w_i benachbart ist oder $v_i = w_i$ gilt.

- Beweisen Sie, dass die Größe der maximalen Clique in $G^{(k)}$ gleich der k -ten Potenz der maximalen Clique in G ist.
- Zeigen Sie, dass für das Problem, eine maximale Clique zu berechnen, ein Approximationschema mit polynomieller Laufzeit gibt (PAS), falls es einen Approximationsalgorithmus mit konstantem Approximationsverhältnis für dieses Problem gibt.

Problem 3: Vertex-Cover

2pt

Ein *Vertex-Cover* eines ungerichteten, einfachen Graphen $G = (V, E)$ ist eine Teilmenge $C \subseteq V$ mit der Eigenschaft, dass für jede Kante $\{u, v\} \in E$ mindestens einer der beiden Knoten u, v in C enthalten ist.

Gegeben seien 2 Approximationsalgorithmen zur Lösung der Problems.

Algorithmus 2 : VERTEX-COVER-APPROX-1(G)

```

1  $C \leftarrow \emptyset$ 
2  $E' \leftarrow E[G]$ 
3 solange  $E' \neq \emptyset$  tue
4    $e \leftarrow \{u, v\}$  beliebige Kante aus  $E'$ 
5    $C \leftarrow C \cup \{u, v\}$ 
6   entferne alle Kanten inzident zu  $u, v$  aus  $E'$ 
7 return  $C$ 

```

Algorithmus 3 : VERTEX-COVER-APPROX-2(G)

```

1  $C \leftarrow \emptyset$ 
2  $E' \leftarrow E[G]$ 
3 solange  $E' \neq \emptyset$  tue
4    $v \leftarrow$  Knoten mit maximalem Grad in  $G' = (V \setminus C, E')$ 
5    $C \leftarrow C \cup \{v\}$ 
6   entferne alle Kanten inzident zu  $v$  aus  $E'$ 
7 return  $C$ 

```

(a) Zeigen Sie, dass Algorithmus 2 ein 2-Approximationsalgorithmus ist.

(b) Zeigen Sie mit Hilfe eines Gegenbeispiels, dass Algorithmus 3 kein 2-Approximationsalgorithmus ist.

(c) Geben Sie einen effizienten Greedy-Algorithmus an, der in linearer (bezogen auf auf $|E|$) Zeit ein optimales Vertex Cover in einem Baum findet.

Problem 4: euklid-TSP-Approximation

1pt

Gegeben sei das bekannte *Problem des euklidischen Handlungsreisenden (e-TSP)*: Gegeben sei eine Menge von n Punkten $P = \{p_1, \dots, p_n\}$ in der Ebene mit Koordinaten (x_i, y_i) für jeden Punkt $p_i \in P$. Gesucht ist die kürzeste Rundtour $T = (p_{i_1}, \dots, p_{i_n})$, die alle Knoten mindestens einmal besucht. Rundtour bedeutet, dass die Tour an dem gleichen (frei wählbaren) Knoten beginnen und enden muss. Der Abstand $d(p_i, p_j)$ zweier Punkte ist definiert durch den euklidischen Abstand der beiden Punkte. Die Länge einer Rundtour $T = (p_{i_1}, \dots, p_{i_n})$ ist $\sum_{j=1}^{n-1} d(p_{i_j}, p_{i_{j+1}}) + d(p_{i_n}, p_{i_1})$.

Algorithmus 4 : E-TSP-APPROX-1(G)

```

1  $T \leftarrow p_1$ 
2 solange  $|T| < |V|$  tue
3    $v, p_{i_j} \leftarrow$  Knoten mit  $d(p_{i_j}, v)$  minimal, mit  $p_{i_j} \in T, v \in V \setminus T$ 
4    $k \leftarrow |T|$ 
5    $T \leftarrow (p_{i_1}, \dots, p_{i_j}, v, p_{i_{j+1}}, \dots, p_{i_k})$  //füge  $v$  zwischen  $p_{i_j}$  und  $p_{i_{j+1}}$  ein
6    $(p_{i_1}, \dots, p_{i_{k+1}}) \leftarrow T$ 
7 return  $T$ 

```

Zeigen Sie, dass Algorithmus 4 ein 2-Approximationsalgorithmus ist.