Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Institut für Theoretische Informatik
Dr. Bernd Gärtner

12. Juni 2003

Theoretische Informatik *Übung 9* *SoSe 2003*

Präsenzaufgabe 1

Wir wollen beweisen, dass ein Problem NP-vollständig ist. Was müssen wir dazu zeigen?

INDEPENDENT SET ist das Entscheidungsproblem, ob ein Graph eine unabhängige Knotenmenge der Grösse k hat. Eine Knotenmenge A ist unabhängig, wenn keine zwei Knoten in A durch eine Kante verbunden sind.

Zeigen Sie die NP-Vollständigkeit von INDEPENDENT SET. Benutzen Sie dazu, dass das Problem CLIQUE NP-vollständig ist, was Sie in Aufgabe 1 zeigen sollen.

Aufgabe 1

Eine *Clique* in einem Graphen ist eine Teilmenge von Knoten, die paarweise durch Kanten miteinander verbunden sind.

Beweisen Sie, dass das Problem CLIQUE (entscheide, ob ein Graph eine Clique der Grösse k hat) NP-vollständig ist.

Hinweis: Zeigen Sie, dass 3-SAT polynomialzeit-reduzierbar ist auf CLIQUE (3-SAT $\leq_P$ CLIQUE). Konstruiere dazu für eine gegebene 3-SAT Formel ϕ mit k Klauseln einen Graphen $G(\phi)$, der genau dann eine Clique der Grösse k hat, wenn ϕ eine erfüllende Belegung besitzt.

Aufgabe 2

Eine Firma plant eine Party für ihre Angestellten. Die Party soll möglichst viel Spass versprechen, weshalb jedem Angestellten ein Spasswert zugewiesen wird, den er zur Party beitragen würde. Andererseits ist die Firma streng hierarchisch organisiert (als Baum, dessen Wurzel der CEO ist), und die Party wäre ein garantierter Misserfolg, wenn ein Angestellter auf seinen direkten Vorgesetzten (Vorgänger im Baum) trifft.

Finden Sie einen effizienten Algorithmus, um eine Gästeliste zu berechnen, die die Summe der Spasswerte der eingeladenen Gäste maximiert, unter der Bedingung, dass kein Angestellter zusammen mit seinem direkten Vorgesetzten eingeladen wird.

Hinweis: Verwenden Sie dynamisches Programmieren!

Aufgabe 3

In der Vorlesung wurde ein Approximationsalgorithmus für TSP mit Dreiecksungleichung gezeigt. Zeigen Sie, dass es für das allgemeine TSP (ohne Dreiecksungleichung) keine Konstante c gibt, für die ein Approximationsalgorithmus mit Güte c existiert, ausser $P = NP$. (Dabei bedeutet Güte c, dass der Algorithmus für jede Instanz des TSP-Problems eine Tour findet, die höchstens c-mal so lang ist wie die optimale Tour.)

Hinweis: Zeigen Sie, dass man mit einem solchen Algorithmus einen Hamiltonschen Kreis in polynomieller Zeit finden könnte.

Abgabe: 19. Juni 2003 in der Vorlesung.