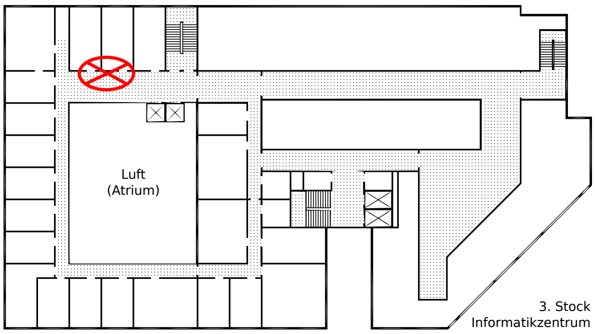
Abteilung Algorithmik **Sommer 2016**
Institut für Betriebssysteme und Rechnerverbund
TU Braunschweig

Prof. Dr. Sándor P. Fekete
Dr. Christian Scheffer

Algorithmen und Datenstrukturen II
Übung 5 vom 22. 06. 2016

Abgabe der Lösungen bis zum Montag, den 04. 07. 2016 um 13:15 im Hausaufgabenrückgabeschrank.

Bitte die Blätter vorne deutlich mit eigenem Namen sowie Matrikel- und Gruppennummer versehen!



Aufgabe 1 (Dynamische Programmierung): Die Firma *Lincoln Consultants* kann gemäß ihres aktuellen Mietvertrags in jedem Monat genau einen von zwei Büroräumen anmieten, die sich an weit voneinander entfernten Standorten, in Seattle und in Miami befinden.

Es soll nun ein optimaler Nutzungsplan für eine Folge von n Monaten im Voraus berechnet werden. Je nach geographischer Verteilung der im Monat i zu beratenden Kunden würden in diesem Monat bei Nutzung des Standorts Seattle Kosten von S_i Dollar anfallen, um alle Kunden zu betreuen, wohingegen bei Nutzung des Standorts Miami M_i Dollar anfallen würden. Der Umzug von einem Standort zu einem anderen Standort kann am Ende eines jeden Monats mit festen Kosten von U Dollar bewerkstelligt werden.

Die Aufgabe besteht nun darin, für eine vorgegebene Folge von Kostenalternativen die minimal notwendigen Gesamtkosten zu bestimmen. Zur Vereinfachung darf angenommen werden, dass es keinen Kostenunterschied für den ersten Monat macht, ob in Miami oder Seattle begonnen wird.

Beispiel: In der nachfolgend angegebenen Situation wäre für $U := 10$ der Wert einer optimalen Lösung $c = 1 + 3 + 10 + 2 + 4 = 20$, da die optimale Lösung in den ersten beiden Monaten das Büro in Seattle nutzt, dann einen Umzug vornimmt und dann das Büro in Miami nutzt.

	Monat 1	Monat 2	Monat 3	Monat 4
Seattle	1	3	20	30
Miami	50	20	2	4

a) Zeige durch Angabe eines Gegenbeispiels, dass der folgende Algorithmus nicht den Wert einer optimalen Lösung bestimmt.

Setze $c := 0$. $\triangleright$ Initialisiere Kosten.
for $i = 1$ to n **do**
    if $(S_i < M_i)$ **then**
        Setze $c := c + S_i$. $\triangleright$ Wähle Seattle.
    else
        Setze $c := c + M_i$. $\triangleright$ Wähle Miami.
    end if
    if (Umzug war notwendig) **then**
        Setze $c := c + U$. $\triangleright$ Addiere ggfs. Umzugskosten.
    end if
    return c.
end for

b) Gib eine Situation an, in der jede optimale Lösung mindestens drei Umzüge erfordert. Begründe kurz Deine Antwort.

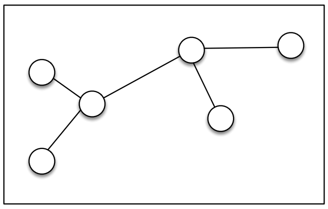
c) Entwirf und analysiere (bzgl. der Laufzeit) einen effizienten (d.h. polynomiellen) Algorithmus, der dynamische Programierung verwendet und der für gegebene Umzugskosten U und eine gegebene Folge von je n Werten M_i und S_i den Wert einer optimalen Lösung des obigen Problems bestimmt.

(3+2+5 Punkte)

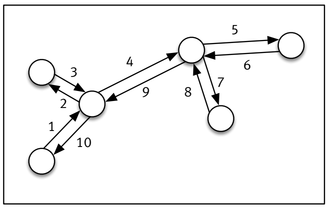
Aufgabe 2 (TSP-Approximation): Betrachte eine n-elementige Punktmenge S in der Ebene. Ein *geometrischer Graph für S* ist ein Graph, dessen Knoten mit den Punkten in S identifiziert werden und in dem alle Kanten mit der euklidischen Distanz der mit ihren Endpunkten identifizierten Punkte aus S gewichtet sind. Ein *minimaler aufspannender Baum für S* ist ein ungerichteter, azyklischer geometrischer Graph für S, der minimales Gesamtgewicht hat. Das *Problem des Handlungsreisenden* besteht darin, für eine gegebene Menge S wie oben einen geometrischen Graph minimalen Gesamtgewichts zu konstruieren, der aus genau einem Zyklus besteht, der jeden Knoten enthält. Dieses Problem ist $\mathcal{NP}$-schwer.

Der folgende, umgangssprachlich gegebene Algorithmus (eine exakte Beschreibung ist für die Bearbeitung dieser Aufgabe nicht notwendig!) bestimmt eine Approximationslösung für dieses Problem, d.h. einen Zyklus wie gefordert, dessen Gesamtgewicht nur um einen konstanten Faktor schlechter ist als die optimale Lösung.

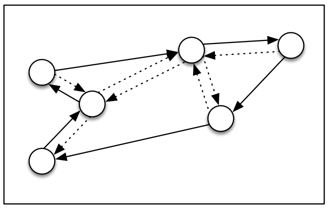
1: Erstelle einen minimalen Spannbaum T für S, wobei die Kantengewichte entsprechend der Distanzen der Endpunkte gegeben seien [dieser Algorithmus darf als gegeben angenommen werden].
2: Ersetze jede Kante $e = (v, w)$ in $E(T)$ durch zwei gerichtete Kanten (v, w) und (w, v) mit dem gleichen Gewicht.
3: Traversiere den so entstehenden Zyklus von einem beliebigen Startknoten aus und überspringe dabei alle Knoten, die bereits besucht wurden.



1. MST



2. Bsp. des Zyklus



3. Ausgabe

Beweise, dass der ausgegeben Zyklus maximal um den Faktor 2 länger ist als die optimale Lösung. **(10 Punkte)**