Universität Karlsruhe (TH)
Lehrstuhl Informatik für Ingenieure und Naturwissenschaftler
Dr. Thomas Worsch
Matthias Schulz

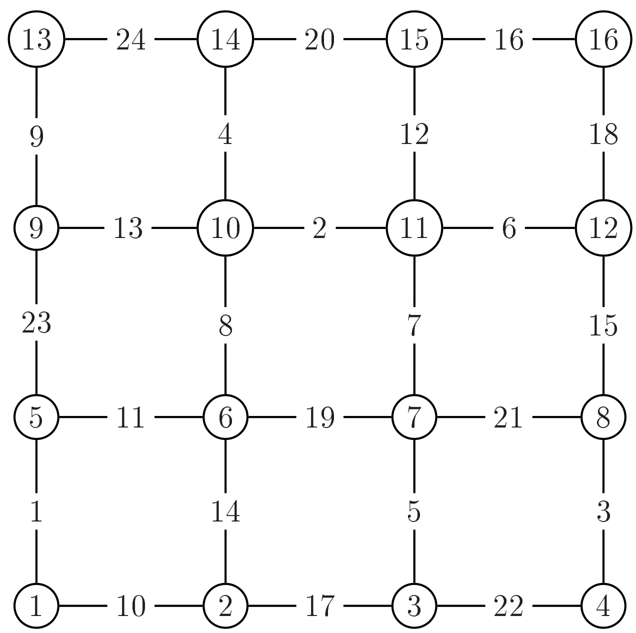
30. April 2008

Einführung in die Informatik

Übungsblatt 3

– Graphentheorie, O-Notation –

Aufgabe 1: (Algorithmus von Prim) Sei $U = (V, E)$ der ungerichtete Graph:



Die Gewichte der Kanten sind durch die Kantenbeschriftungen gegeben.

a) Berechnen Sie mit Hilfe des Algorithmus von Prim einen minimalen spannenden Baum! Beginnen Sie mit Knoten 1 und Nummerieren Sie die Kanten des Baumes in der Reihenfolge, in der Sie hinzugefügt wurden. [`autotool`: MST-Gewichte. Probieren Sie auch die `autotool`-Aufgabe MST!]

b) Welche Zeitkomplexität hat der Algorithmus von Prim in der O-Notation bei Graphen mit n Knoten und m Kanten? Welche Voraussetzungen haben Sie benutzt?

Aufgabe 2: (Bisketionsweite, Durchmesser)
Gegeben sei der ungerichtete Graph $U = (V, E)$ mit $V = \{1, \ldots, 6\}$ und der Adjazenzmatrix

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

(Siehe Übungsblatt 2, Aufgabe 2)
Bestimmen Sie Durchmesser und Bisektionsweite von U. Geben Sie für die Bisektionsweite die Menge der Kanten an, die Sie entfernt haben und erklären Sie, wie Sie den Durchmesser bestimmt haben! [`autotool`: Bisekt-Direct. Bestenliste: Kleinste Kantenzahl. Probieren Sie auch die `autotool`-Aufgabe Bisekt-Quiz!]

Aufgabe 3: (O-Notation)

a) Zeigen oder widerlegen Sie: $n \in O(n^2)$.

b) Zeigen oder widerlegen Sie: $O(n^2) \subseteq O(n^3)$.

c) Zeigen oder widerlegen Sie: $O(f) = O(g) \Rightarrow \Theta(f) = \Theta(g)$.

d) Zeigen oder widerlegen Sie: $O(2^n) \subseteq O((n+1)^{-1} \cdot 2^n)$

e) Zeigen oder widerlegen Sie: $O(f(n)) \cap \Omega(f(n)) = \Theta(f(n))$.

f) Zeigen oder widerlegen Sie: $O(n^2) \subseteq \Theta(n^3)$.

g) Zeigen oder widerlegen Sie: Falls $f \in O(g)$ und $g \in O(f)$ gilt, so folgt $O(f) = O(g)$.

Aufgabe 4: (Komplexität)
Es seien zwei Algorithmen A_1 und A_2 zur Lösung desselben Problems (z.B. Sortieren von n Zahlen) gegeben. Die Laufzeit von A_1 bei sei durch die Funktion $L_1(n) = 4L_1(n/2) + 2$ und die Laufzeit von A_2 durch die Funktion $L_2(n) = 2L_2(n/2) + n$ gegeben. Welcher Algorithmus ist zu bevorzugen? Begründen Sie Ihre Entscheidung!
Abgabe bis zum **7. Mai 2008** in der Vorlesung oder im Tutorium.
Falls Sie eine Bearbeitung abgeben möchten, geben Sie bitte den Namen Ihres Tutors und Ihre Übungsgruppe an.