Technische Universität München
Institut für Informatik
Prof. Dr. Ernst W. Mayr

WS 1998/99
Aufgabenblatt 4
30. November 1998

Praktikum Algorithmen-Entwurf

(Abgabetermin: Montag, den 7. Dezember 1998, 24.^{00}Uhr)

Kürzeste Pfade

Aufgabe 1 Algorithmus von Dijkstra

(a) Gegeben sei ein gerichteter Graph $G = (V, E)$ mit ganzzahligen, positiven Kantengewichten. Implementieren und animieren Sie den Algorithmus von Dijkstra so, daß die kürzesten Pfade von einem vom Benutzer gewählten Startknoten zu allen davon erreichbaren Knoten berechnet und dargestellt werden. Bei jedem Knoten soll die Länge des berechneten kürzesten Pfades zu diesem Knoten als Knotenlabel angezeigt werden.
Während der Algorithmus abläuft, sollen zu jeder Zeit die schon erledigten Knoten und die in der Priority-Queue befindlichen Knoten farblich gekennzeichnet sein. Ebenso sollen die bereits gefundenen kürzesten Pfade gut erkennbar sein. Für jeden Knoten in der Priority-Queue soll zusätzlich diejenige Kante markiert werden, über die er momentan am günstigsten von einem schon erledigten Knoten aus erreicht werden kann.

(b) Gegeben sei ein gerichteter Graph $G = (V, E)$, der ein Kommunikationsnetzwerk modelliert, in dem Verbindungen mit einer gewissen Wahrscheinlichkeit ausfallen können. Jeder Kante $e \in E$ ist eine Ausfallwahrscheinlichkeit p_e mit $0 \leq p_e \leq 1$ zugeordnet. Die Ausfälle verschiedener Kanten seien unabhängig voneinander. Ein Pfad fällt aus, wenn mindestens eine Kante auf dem Pfad ausfällt. Implementieren und animieren Sie einen effizienten Algorithmus, der für zwei vom Benutzer ausgewählte Knoten s und t einen gerichteten Pfad P von s nach t berechnet und anzeigt, so daß die Ausfallwahrscheinlichkeit des Pfades unter allen Pfaden von s nach t minimal ist. Die Ausfallwahrscheinlichkeit von P soll auch angezeigt werden.

Aufgabe 2 Azyklische Graphen

Gegeben sei ein gerichteter Graph $G = (V, E)$ mit beliebigen ganzzahligen Kantengewichten, der keine Kreise enthält. Implementieren und animieren Sie einen Algorithmus, der *in linearer Zeit*, d.h. Zeit $O(|V| + |E|)$ statt $O(|V| \log |V| + |E|)$ wie beim Algorithmus von Dijkstra, die kürzesten Pfade von einem vom Benutzer gewählten Startknoten v zu allen von v erreichbaren Knoten berechnet. Erweitern Sie das Programm so, daß wahlweise die kürzesten oder die längsten Pfade berechnet werden können!

Hinweise

Passende Eingabegraphen sind `pos1.gw` und `pos2.gw` für Aufgabe 1a, `fail1.gw` und `fail2.gw` für Aufgabe 1b und `acyc1.gw` und `acyc2.gw` für Aufgabe 2. Die Graphen stehen an den üblichen Orten zur Verfügung. Die Gewichte der Kanten sind wieder als Strings im User-Label gespeichert und sollten in ein `edge_array<int>` (Aufgabe 1a und 2) bzw. ein `edge_array<double>` (Aufgabe 1b) eingelesen werden.