Technische Universität München
Institut für Informatik
Prof. Dr. Angelika Steger

SS 2003
Aufgabenblatt 5
26.05.2003

Praktikum Diskrete Optimierung

Abgabetermin: Montag, den 16.06.2003, 14.00 Uhr

Aufgabe 6 Heuristiken: Matchings in gewichteten bipartiten Graphen

Gegeben sei ein vollständiger, bipartiter Graph. Die Kantengewichte seien zufällig aus dem Intervall $[0;1]$ gewählt (nicht unbedingt Gleichverteilung!). Implementieren Sie eine heuristische Suche nach einem gewichtsmaximalen Matching in ungerichteten gewichteten bipartiten Graphen. Verwenden Sie für Ihre Heuristik folgende allgemeine Vorgehensweise:

1. Versuchen Sie den vollständigen Graphen zunächst weitgehend "auszudünnen", d.h. möglichst viele Kanten aus dem Graphen zu entfernen, die wahrscheinlich nicht zu einem gewichtsmaximalen Matching gehören können.

2. Verwenden Sie dann den Algorithmus basierend auf der Suche nach geeigneten augmentierenden Pfaden mit Hilfe des Dijkstra-Algorithmus, den Sie bereits von Übungsblatt 4, Aufgabe 5, kennen. Da die Laufzeit dieses Algorithmus mit $O\left(|V| \cdot (|E| + |V| \log |V|)\right)$ wesentlich von der Anzahl vorhandener Kanten abhängt, sollte dieser auf einem stark "ausgedünnten" Graphen schnell eine Lösung finden können.

Ihr C++-Programm soll den Graphen von der *Standardeingabe* lesen und ein beliebiges gewichtsmaximales Matching auf der *Standardausgabe* ausgeben.
Wichtiger Hinweis: Geben Sie zusätzlich zu Ihrem Programm bitte eine Textdatei mit einem ausführlichen Kommentar zu Ihrer verwendeten Heuristik ab. Beschreiben Sie darin detailliert Ihre Vorgehensweise. Bereiten Sie zusätzlich einen kurzen, *maximal* 3 minütigen **Vortrag** zu Ihrer Heuristik vor – die *Anwesenheit* bei der nächsten Praktikumsbesprechung ist deshalb für alle Gruppen verpflichtend.

Wichtiger Hinweis: Halten Sie das im folgenden angegebene Eingabe- und Ausgabeformat bitte genau (keinerlei zusätzliche Leerzeichen, Zeilenumbrüche etc.) ein! Die abgegebenen Programme werden u.a. auch mit Hilfe eines automatisierten Verfahrens auf verschiedenen Eingaben getestet.

Eingabeformat

Als Eingabe erhält Ihr Programm zunächst eine Zeile mit der Knotenanzahl n_1 der ersten Partition getrennt durch ein Whitespace von der Knotenanzahl n_2 der zweiten Partition des Graphen. Alle Knoten in der ersten Partition werden durch die Zahlen zwischen 0 und $n_1 - 1$, die der zweiten Partition durch Zahlen zwischen n_1 und $n_1 + n_2 - 1$ repräsentiert. In den darauffolgenden

Zeilen finden sich für jede Kante des Graphen jeweils drei durch Whitespaces getrennte Werte: zwei Integer-Werte für das Knotenpaar und ein Double-Wert (> 0) für das Kantengewicht. Die Kanten des Graphen werden also wie in der letzten Aufgabe als Paare von Knoten angegeben, wobei jede (ungerichtete) Kante in der Auflistung nur einmal vorkommt und nur Kanten vorhanden sind, die Knoten aus den zwei verschiedenen Partitionen verbinden.

Beispieleingabe

```
3 3
0 3 0.1
0 4 1.0
0 5 0.8
1 3 0.5
1 4 0.7
1 5 0.3
2 3 1.0
2 4 0.2
2 5 0.6
```

Ausgabeformat

Ihr Programm soll das gefundene (fast) gewichtsmaximale Matching im Eingabegraphen in einer beliebigen Reihenfolge ausgeben. Für jede Kante im Matching sollen jeweils die inzidenten Knoten durch ein Whitespace getrennt ausgegeben werden. Hierbei soll jede Kante *genau einmal* ausgegeben werden. Geben sie dabei *nicht* das jeweilige Kantengewicht aus.

Beispielausgabe

```
0 4
1 5
2 3
```

Weitere Beispieleingaben stehen unter
`http://www14.in.tum.de/lehre/2003SS/optprak/data.html`
zur Verfügung.

Zeitlimit und Abgabe

Für diese Aufgabe wird es kein festes Zeitlimit geben. Selbstverständlich sollte eine Heuristik wesentlich schneller sein als eine exakte Lösung, langsamere Lösungen werden nicht akzeptiert. Wir werden die Aufgaben nach Geschwindigkeit und Güte der Lösung bewerten.
Die Abgabe der Lösung muß bis Montag, 16.06.2003, 14.^{00} Uhr, per E-Mail an die Adresse `optprak@in.tum.de` erfolgen.