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

SS 2003
Aufgabenblatt 4
12.05.2003

Praktikum Diskrete Optimierung

Abgabetermin: Montag, den 26.05.2003, $14.^{00}$ Uhr

Aufgabe 5 Matchings in gewichteten bipartiten Graphen

Implementieren Sie die Suche nach einem gewichtsmaximalen Matching in ungerichteten gewichteten bipartiten Graphen. Verwenden Sie nun einen Algorithmus basierend auf der Suche nach geeigneten augmentierenden Pfaden mit Hilfe des Dijkstra-Algorithmus. Ihr Algorithmus sollte, unter Verwendung von Fibonacci-Heaps, eine Laufzeit von $O\left(|V| \cdot (|E| + |V| \log |V|)\right)$ besitzen (näheres hierzu finden Sie wie üblich in der angegebenen Literatur).
Ihr C++-Programm soll den Graphen von der *Standardeingabe* lesen und ein beliebiges gewichtsmaximales Matching auf der *Standardausgabe* ausgeben.
Verwenden Sie Fibonacci-Heaps zur Implementierung der Priority Queue, die im Algorithmus von Dijkstra benötigt wird. Eine Implementierung der Fibonacci-Heaps steht auf unserer Website
`http://www14.in.tum.de/lehre/2003SS/optprak/data.html`
zum Download bereit. Auch andere Implementierungen der Fibonacci-Heaps sind bei dieser Aufgabe zugelassen, solange Sie keine zusätzlichen Bibliotheken erfordern. Sie können die Fibonacci-Heaps jedoch auch selbst implementieren (sehr aufwendig!).

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

```
6 8
0 6 10.0
0 7 5.0
0 8 7.2
1 9 4.67
2 10 1.8
3 10 8.7
3 11 6.4
4 11 2.3
4 12 2.4
5 12 9.74
```

Ausgabeformat

Ihr Programm soll das gefundene 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 6
1 9
3 10
4 11
5 12
```

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

Zeitlimit und Abgabe

Pro Testeingabe stehen maximal 30 Sek. Laufzeit zur Verfügung.
Die Abgabe der Lösung muß bis Montag, 26.05.2003, $14.^{00}$ Uhr, per E-Mail an die Adresse `optprak@in.tum.de` erfolgen.

Literaturhinweise

- Johan Hastad. Notes for the course advanced algorithms. KTH Nada, Sweden, January 2000. Verfügbar als Postscript-Datei unter:

  `http://www.nada.kth.se/~johanh/`