

Algorithmische Mathematik I

Programmieraufgabe 2

Kruskals Algorithmus

Implementieren Sie Kruskals Algorithmus zur Berechnung eines minimalen aufspannenden Baumes in einem zusammenhängenden Graphen mit einer Laufzeit von $O(m \log n)$.

Einlesen der Daten: Dem Programm muss beim Aufruf der Name einer Datei übergeben werden. Ein Aufruf hat also die Form

<programmname> <dateiname>

Eine gültige Datei, die einen Graphen beschreibt, hat das folgende Format:

```
Knotenanzahl  
Kantenzahl  
Knoten0a Knoten0b Gewicht0  
Knoten1a Knoten1b Gewicht1  
...
```

Die Einträge der Datei sind ausschließlich ganze Zahlen. Sie können voraussetzen, dass die Summe der Absolutbeträge aller Zahlen in der Eingabe kleiner als 2^{31} ist. In den ersten beiden Zeilen steht jeweils eine einzelne natürliche Zahl (größer als 0), welche in der ersten Zeile die Anzahl der Knoten und in der zweiten die Anzahl der Kanten angibt. Jede weitere Zeile spezifiziert genau eine Kante. Die ersten beiden Einträge einer Zeile sind zwei verschiedene nichtnegative ganze Zahlen, welche die Nummern der Endknoten der Kante sind. Dabei nehmen wir an, dass, wenn wir n Knoten haben, die Knoten von 0 bis $n - 1$ durchnummeriert sind. Der dritte Eintrag in der Zeile ist eine ganze Zahl, die das Gewicht der Kante bezeichnet. Der Index einer jeden Kante ist durch ihre Zeilennummer in der Eingabedatei gegeben: Zeile i kodiert die Kante mit Index $i - 3$ (für $i = 3, \dots, m + 2$, wobei m die Zahl der Kanten sei). Parallele Kanten können vorkommen.

Ausgabeformat: Das Programm muss in der ersten Zeile der Ausgabe das Gewicht des berechneten Baums ausgeben. Die weiteren Zeilen müssen jeweils genau einen Index einer Kante des Baumes enthalten. Jeder Index einer Baumkante muss dabei in genau einer Zeile vorkommen, und die Indizes müssen aufsteigend sortiert sein.

Beispiel: Eine Eingabedatei für einen Graphen mit vier Knoten und sechs Kanten kann so aussehen:

```
4
6
0 3 15
1 3 25
1 2 -12
1 2 21
2 3 18
0 1 10
```

Die Ausgabe des Programms muss dann so aussehen:

```
13
0
2
5
```

Das Programm muss in C++ geschrieben sein. Für die Sortierung dürfen Sie `std::sort` verwenden. Die Laufzeit dieser Sortier Routinen soll hier nicht betrachtet werden. Andere Algorithmen aus externen Bibliotheken dürfen nicht verwendet werden.

Weitere Hinweise, Instanzen mit Lösungen und ein Beispielprogramm, mit dem sich Graphen einlesen lassen, befinden sich auf der Seite

http://www.or.uni-bonn.de/lectures/ws08/alma1_08_uebung.html

Auf dieser Seite finden Sie auch fünf Instanzen, bei denen keine Lösungen angegeben sind. Lassen Sie Ihr Programm auch auf diesen Instanzen laufen, und **geben Sie die Längen der berechneten Bäume zusammen mit Ihrem Programm ab.**

Bitte beachten Sie: Die **Klausurzulassung** erhalten Sie nur bei erfolgreicher Bearbeitung dieser Programmieraufgabe.

Abgabe: Bis spätestens Freitag, den 30.1.2009, im CIP-Pool, Wegelerstraße 6. Tragen Sie sich bitte rechtzeitig in die dort aushängenden Listen ein.