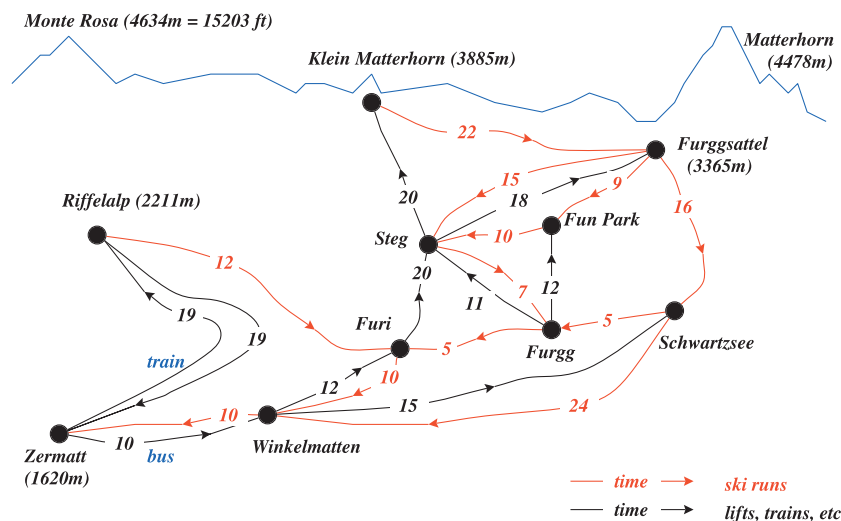


Einführung in die Diskrete Mathematik

9. Übung

- Man betrachte eine Verallgemeinerung des MINIMUM-COST-FLOW-PROBLEMS, bei der unendliche Kapazitäten erlaubt sind (d.h. $u(e) = \infty$ für manche Kanten e). Eine Instanz (G, u, b, c) heißt *unbeschränkt*, wenn es für jedes $\gamma \in \mathbb{R}$ einen b -Fluss f in (G, u) gibt mit $c(f) < \gamma$.
 - Man zeige, dass eine Instanz genau dann unbeschränkt ist, wenn es einen b -Fluss in (G, u) gibt und ein negativer Kreis existiert, dessen Kanten alle unendliche Kapazität haben.
 - Man zeige, wie man in $O(n^3 + m)$ -Zeit entscheiden kann, ob eine Instanz unbeschränkt ist.
 - Man zeige, dass in einer nicht unbeschränkten Instanz jede unendliche Kapazität auf äquivalente Weise durch eine endliche Kapazität ersetzt werden kann. (2+1+2 Punkte)
- Unten sehen Sie einen stark vereinfachten Plan der Skipisten in Zermatt. Die Pisten selbst sind in rot dargestellt, Skilifte und andere Transportmöglichkeiten in schwarz. Was ist die kürzeste Zeit, in der man, wenn man in Zermatt beginnt und endet, alle Skipisten abfahren kann? Zeigen Sie (auf möglichst einfache Weise), dass Ihre Lösung tatsächlich optimal ist. (5 Punkte)



- Implementieren Sie den SUKZESSIVE-KÜRZESTE-WEGE-ALGORITHMUS, um das ZUORDNUNGSPROBLEM zu lösen. Das Programm soll Laufzeit $O(n^3)$ haben, wobei $n = |V(G)|$ sei. (10 Punkte)

Abgabe: Dienstag, den 17.12.2013, vor der Vorlesung.

Hinweise zur Programmierübung:

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

Knoten0a Knoten0b Kosten0

Knoten1a Knoten1b Kosten1

...

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 der ersten Zeile steht eine einzelne positive gerade Zahl n , welche die Anzahl der Knoten angibt. Die Knoten werden von 0 bis $n-1$ durchnummeriert. Jede folgende Zeile kodiert genau eine Kante. Die ersten beiden Einträge einer Zeile sind die Nummern der Endknoten der Kante (wobei die Kante vom jeweils ersten angegebenen Knoten zum zweiten gerichtet sei). Die erste Nummer liegt dabei in $\{0, \dots, \frac{n}{2} - 1\}$ und die zweite in $\{\frac{n}{2}, \dots, n-1\}$. Der dritte Eintrag in der Zeile gibt die Kosten der Kante an.

Parallele Kanten kommen in den Instanzen nicht vor.

Ausgabeformat: Das Programm muss in der ersten Zeile der Ausgabe die Kosten der berechneten Lösung ausgeben. Danach folgen $\frac{n}{2}$ weitere Zeilen, die jeweils genau zwei Zahlen enthalten und eine Kante der berechneten Lösung kodieren. Die Nummern in der jeweiligen Zeile geben dabei die Nummern der Endknoten an (wobei die kleinere Nummer zuerst stehen soll).

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

```
6
2 5 37
0 3 12
0 4 -14
1 4 8
2 3 4
1 5 2
```

Die Ausgabe der Programms kann dann so aussehen (die letzten drei Zeilen können anders sortiert sein):

```
-8
0 4
2 3
1 5
```

Das Programm muss in C++ geschrieben sein. Es muss korrekt arbeiten und ohne Fehlermeldung kompiliert werden können. Der Code muss auf einem gängigen Linuxsystem funktionieren. Algorithmen aus externen Bibliotheken dürfen nicht verwendet werden.

Zur Berechnung der kürzesten Wege soll Dijkstras Algorithmus benutzt werden. Dabei genügt eine Implementierung mit Laufzeit $O(n^2)$.

Abgabe: Der Quelltext des Programms muss bis 17. Dezember 2013, 16:15 Uhr per E-Mail beim jeweiligen Tutor eingegangen sein. Außerdem ist bis zu diesem Zeitpunkt ein Ausdruck des Quelltextes zusammen mit den Theorieaufgaben abzugeben.

Testinstanzen befinden sich ab dem 11.12.2013 auf der Seite

http://www.or.uni-bonn.de/lectures/ws13/edm_13_uebung.html