# Protokoll Programmierbeispiel 3 - Graphen Suche

***Aufgabenstellung:***

*Gegeben ist ein Graph mit gewichteten Kanten. Entwickeln und implementieren Sie einen Algorithmus, der den kostengünstigsten Weg zwischen zwei Knoten findet. Der zu durchsuchende Graph ist durch eine Textdatei mit folgendem Aufbau gegeben. Jede Zeile beginnt mit dem Namen einer Linie im Wiener Verkehrsnetz (nur U-Bahn und Straßenbahn), gefolgt von einem Doppelpunkt. Danach folgen abwechseln ein String unter Anführungszeichen (Stationsname) und eine Zahl (Fahrzeit zur nächsten Station=Kosten der Kante). Es kann auch zyklische Linien geben (d.h. Stationen können mehrfach angefahren werden).*

*Ihr Programm soll zuerst den Graphen aus dem Input-File lesen und in eine geeignete Datenstruktur speichern. Anschließend wird der Algorithmus zur Suche des kürzesten Pfades vom Start zum Ziel ausgeführt und der gefundene Pfad übersichtlich ausgegeben. Das Output-Format ist nicht fix vorgegeben, soll aber zumindest darstellen*

*•welche Kanten der Reihe nachdurchlaufen werden und welche Linien dabei jeweils verwendet werden  
•wo umgestiegen werden muss  
•und wie hoch die Gesamtkosten (=Fahrzeit) sind.*

### Definition Datenstruktur:

Gewählt wurde in eine Map, die mithilfe von String-Indizes unterschiedliche Stationen speichern kann. Die Stationen werden in Strukturen gespeichert, bei denen jeweils Name, ein Array mit Verbindungen zu anderen Knoten, das vorherige Element und die Gesamtkosten. Die letzten zwei erwähnten Attribute der Struktur sind relevant für den Dijkstra Algorithmus und zur Pfadfindung zwischen zwei Stationen. Für die Verbindung wurde ebenfalls eine Struktur implementiert, die jeweils die Kosten, die Metro-Linie und die Station speichert. Die Station in der Verbindungsstruktur wird dabei als Pointer gespeichert.

### Suchalgorithmus

Für die Suche im Graphen wurde der Dijkstra Algorithmus implementiert. Er gehört zu den Greedy-Algorithmen, die eine spezielle Klasse von Algorithmen in der Informatik bilden. Besonders an gierigen Algorithmen ist, dass sie den Folgezustand so wählen, dass das beste Ergebnis zur entsprechenden Wahl eines Startpunktes gefunden wird. Sie lösen viele Probleme nicht optimal, sind jedoch in der Regel schnell.

### Laufzeit

Die Laufzeit für den Dijkstra ist wie folgt definiert:

Wobei E die Anzahl der Kanten und V die Anzahl der Knoten ist. Jeder Knoten mit (K-1) Knoten verbunden werden. (K-1) wird durch E repräsentiert. Das Finden und Aktualisieren der Gewichte jeder Knoten erfolgt in einem binären Min-Heap und hat damit eine Laufzeit von O(log(K)) +O (1). Da dies für jede Kante in E geschehen muss, beträgt die Laufzeit wie in der oben genannten Formel.