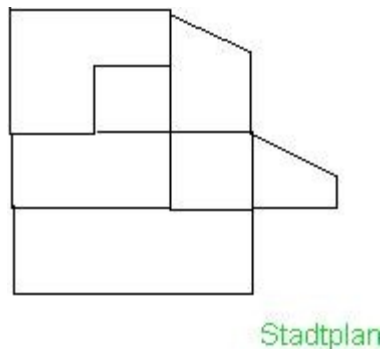


Plan für die Müllabfuhr

Sucht man in einem vorgegebenen Stadtplan einen Weg für die Müllabfuhr, dann geht es zwar zunächst darum, dass überhaupt alle Straßenabschnitte angefahren werden, bei einer Optimierung wird man aber nach Möglichkeit fordern, dass jeder Weg genau einmal befahren wird¹.

Wir betrachten den folgenden Stadtplan:



Welche Überlegungen müssen wir anstellen?

- lösbar? - Kriterien ?
- gibt es eine Optimum ?
- Datenstruktur ?
- Suchverfahren ?
- ... im Zusammenhang mit dem Umfang des Problems

Schreiben Sie das Programm!

¹ Die weitere Forderung, dass unser Müllfahrzeug keine Sprünge macht, der Graph des Stadtplans also zusammenhängend ist, sollte eigentlich trivial sein.

Zu den Überlegungen:

Es handelt sich um das Problem des Eulerschen Weges. Damit gilt, dass die Aufgabe, jeden Abschnitt genau einmal zu befahren, genau dann lösbar ist, wenn die Eulersche Bedingung erfüllt ist:

Ein zusammenhängender ungerichteter Graph ist **genau dann** ein Eulerscher Graph, wenn die Zahl der zugehörigen Kanten jedes Knotens gerade ist.

Das ist der angegebene sicher nicht!

Natürlich gibt es eine Optimum. Die Anzahl der möglichen vollständigen Wege ist beschränkt, wenn wir zu jedem höchstens einmal eine Wiederholung zulassen. Zu jedem der Wege gibt es eine Bewertungszahl und zwar eine positive. Die Menge aller Bewertungszahlen ist endlich und hat damit ein Minimum.

Als Datenstruktur verwenden wir eine bewertete Kantenliste.

Das Suchproblem ist bei einem echten Eulerschen Graphen leider keines, da es ein direktes Verfahren gibt. Siehe dazu den html-Text.

Programm:

Der im Text beschriebene Beweis von Turau lautet:

Beweis. Es sei zunächst W ein geschlossener Kantenzug von G , welcher jede Kante genau einmal enthält. Es sei e eine beliebige Ecke von G . Jedes Vorkommen von e auf W trägt 2 zum Eckengrad von e bei. Da alle Kanten genau einmal auf W vorkommen, muss $g(e)$ eine gerade Zahl sein.

Es sei nun G ein zusammenhängender ungerichteter Graph, so dass der Eckengrad jeder Ecke gerade ist. Nun starte man einen Kantenzug an einer beliebigen Ecke des Graphen. Man durchlaufe den Graphen und entferne jede verwendete Kante. Bei jedem Durchlaufen einer Ecke sinkt der Eckengrad um 2. Da alle Ecken geraden Eckengrad haben, muss der Kantenzug bei der Startecke enden. Wurden alle Kanten schon verwendet, so handelt es sich um einen Eulerschen Graphen. Anderenfalls gibt es eine Ecke auf dem Kantenzug, welche noch zu unbenutzten Kanten inzident ist. Von dieser Ecke wird erneut ein Kantenzug auf die gleiche Art erzeugt. Dieser Kantenzug muss ebenfalls an seiner Startecke enden. Die beiden Kantenzüge werden nun zu einem Kantenzug verschmolzen. Man beachte, dass die Ecken in dem Restgraphen alle geraden Eckengrad haben. Auf diese Art wird ein geschlossener Kantenzug erzeugt. Dieser enthält jede Kante genau einmal. Somit ist G ein Eulerscher Graph.

Der zweite Teil des Beweises lässt sich leicht in einen Algorithmus mit Laufzeit $O(m)$ umsetzen. ...

Richtig interessant wird es also erst, wenn man keinen echten Eulerschen Graphen hat und daher einige Wege doppelt gegangen werden müssen. Welche muss man dafür auswählen und welche kann man auswählen ?

