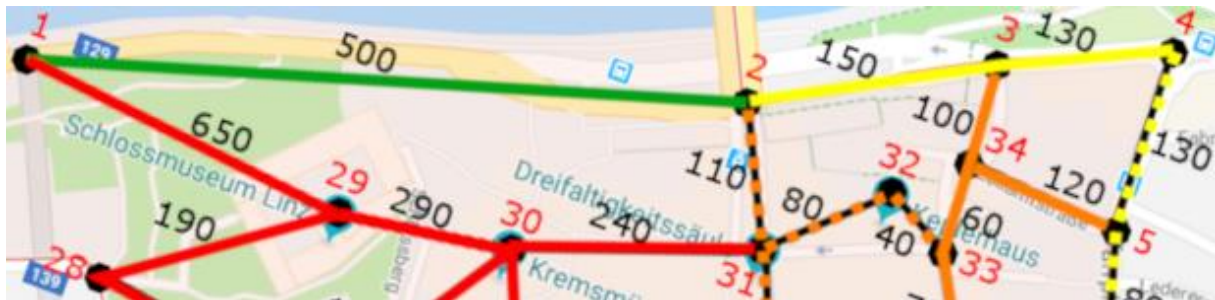


# Flussproblem v1.0.1 (3 EH)

## Szenario

Gegeben ist der Graph eines **Straßennetz**: Die **Knoten** stehen dabei für die Kreuzungen und die **Kanten** für die Straßen dazwischen. Die **Kantengewichte** beschreiben die **Kapazität an Fahrzeugen**, die die jeweilige Straße in einer Stunde passieren können. Ziel ist es einen Algorithmus zu programmieren, welcher die **maximal mögliche Menge an Fahrzeugen** (sog. *Fluss*) zwischen zwei beliebigen Kreuzungen findet, um damit **Engpässe im Straßennetz** zu finden.

Hier spielen nun auch die **Farben der Straßen** in der Linzer Altstadt eine Rolle: Grüne Straßen erlauben 2000 Fahrzeuge, gelbe Straßen 1000, orange Straßen 500 und rote Straße 250.



## Aufgabe (3 EH)

- Arbeiten an dem **Projekt** weiter, in dem ihr das *Suchproblem* programmiert habt.
- Lest die Datei *Linz\_Flussproblem.csv* ein.
- Programmiert und testet: **Graph#determineMaximumFlow(int, int)**
  - Bestimmt den maximalen Fluss zwischen den beiden als Index übergebenen Knoten.
  - Schreibt mindestens einen Unit-Test, behandelt auch ungültige Parameter.
- Programmiert und testet: **Graph#determineBottlenecks(int, int)**
  - Bestimmt alle Kanten im Graph, bei denen der maximale Fluss zwischen den beiden als Index übergebenen Knoten dazu führt, dass ihre Kapazität erschöpft ist.
  - Schreibt mindestens einen Unit-Test, behandelt auch ungültige Parameter.
- Programmiert und testet: **Edge#getFirstNodeId()** und **Edge#getSecondNodeId()**
  - Schreibt mindestens einen Unit-Test.
- Überlegt gemeinsam, wie man den zweitgrößten Fluss zwischen zwei beliebigen Kreuzungen finden kann.

## Abgabe

Gebt das komplette Projekt (inkl. *Suchproblem*) im Moodle ab.