WISSEN
TECHNIK
LEIDENSCHAFT

Lab programme 1: Routing [Tutorial]

Navigation Systems

WS 2025/26

Aim

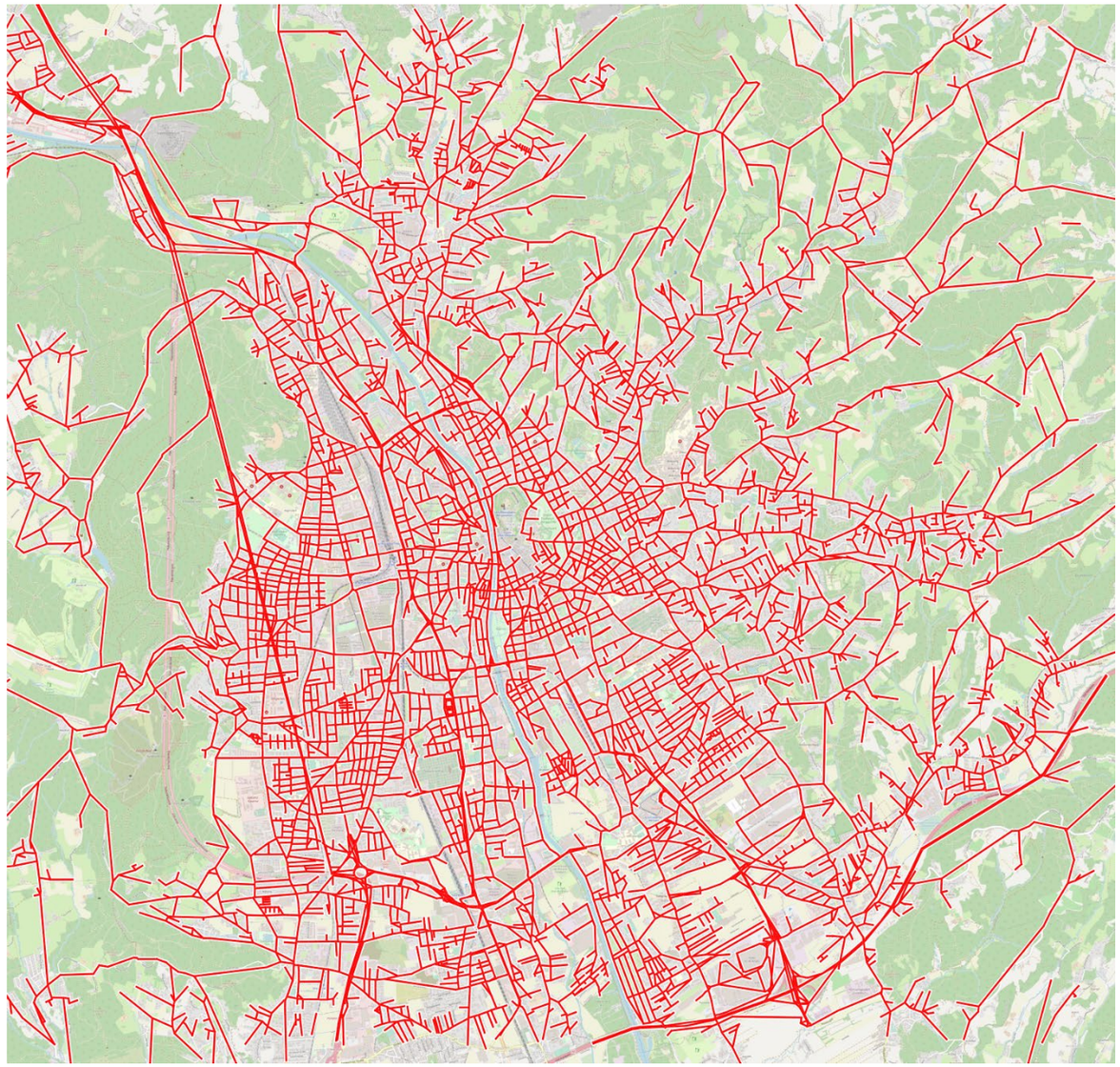
- Generate optimal routes
- Dijkstra algorithm

- Start node: place where one person of your group lives
  - Visually or calculated (DataFrame.idxmin(...))

- End nodes:
  - Basilika Mariatrost (node 9328)
  - Schloss Eggenberg (node 6031)
  - Shopping-Center Murpark (node 8543)

Given data

- nodepl.txt
  - Coordinates of nodes (φ and λ in degrees)
- nodelist.txt
  - Adjacency list of nodes
- arclist.txt
  - Adjacency list of arcs
  - Columns: time, distance, speed limit, clazz, flags



Tasks

- Cost functions
  - Time
  - Distance

- Additionally, choose between:
- 2.1 Self-developed cost-function for Dijkstra’s algorithm
- 2.2 Heuristic A* algorithm

Deliverables

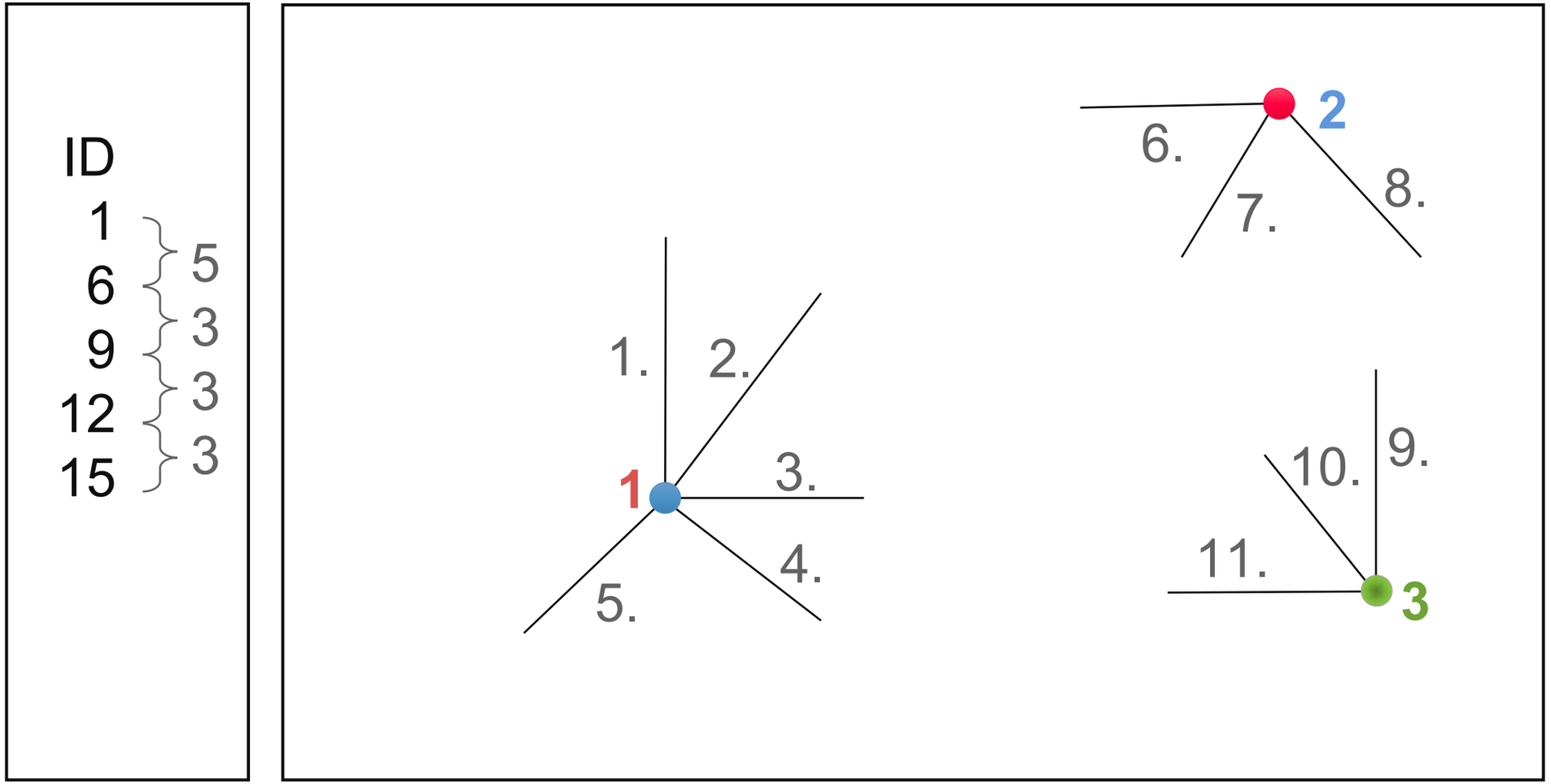
- Source code
- Presentation
  - Visualization of optimal routes (time & distance)
  - Evaluation of routes
  - Self-made cost function *or* A*-algorithm
  - 5 minutes video with audio commentary

Dijkstra Algorithm

Dijkstra algorithm

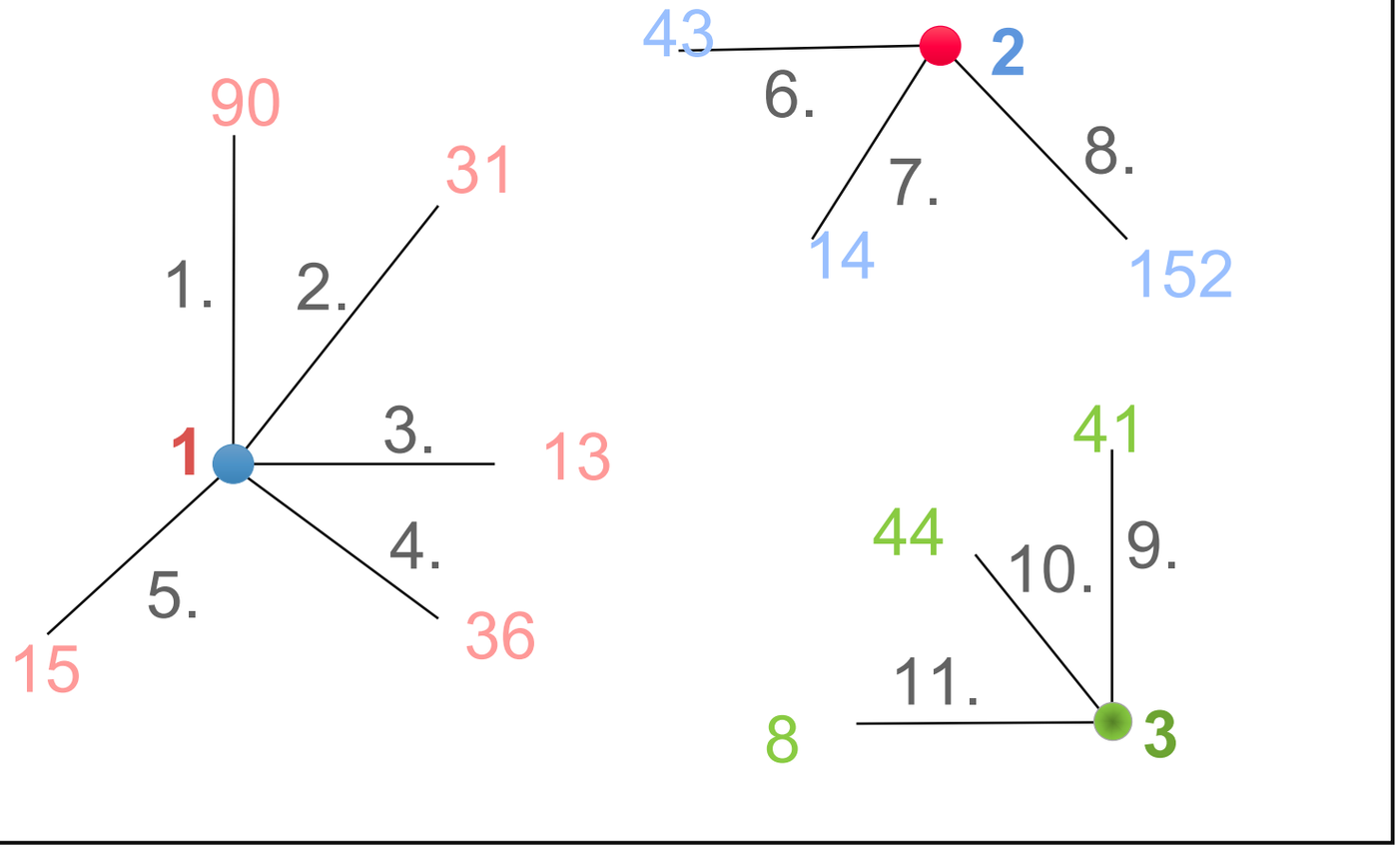
- Shortest path from one node to all other nodes
- Assign temporary or permanent labels during search process
- Predecessor list: finally represents the nodes of the shortest path

Adjacency list of nodes



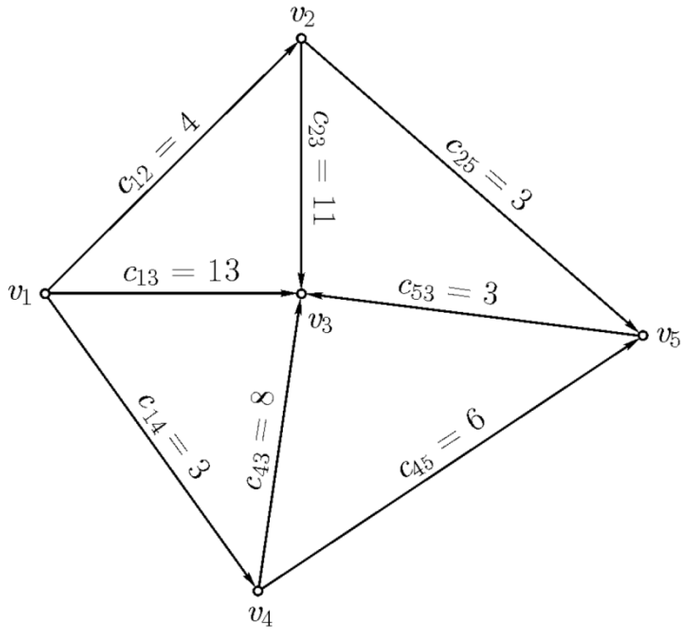
Adjacency list of arcs

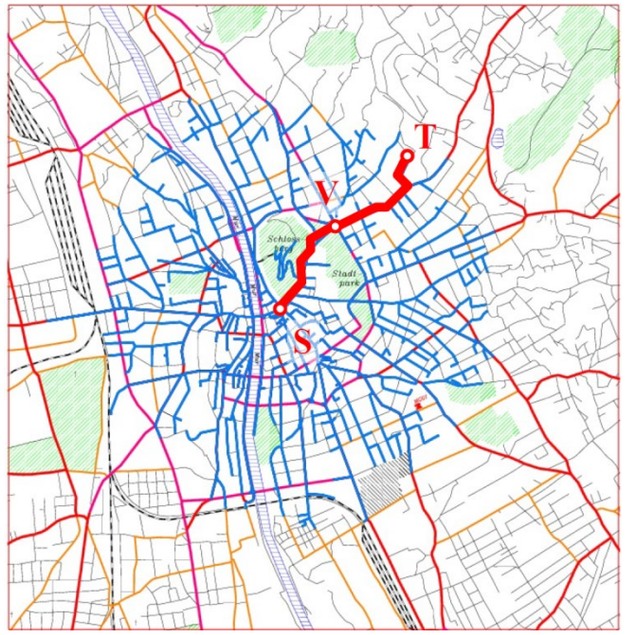
v_j	c_{ij} [m]
90	63.20
31	78.77
13	40.25
36	78.26
15	81.44
43	34.78
14	81.44
152	81.44
41	77.18
44	11.90
8	45.89



Dijkstra algorithm

- Check with small amount of data
  - Example from lecture





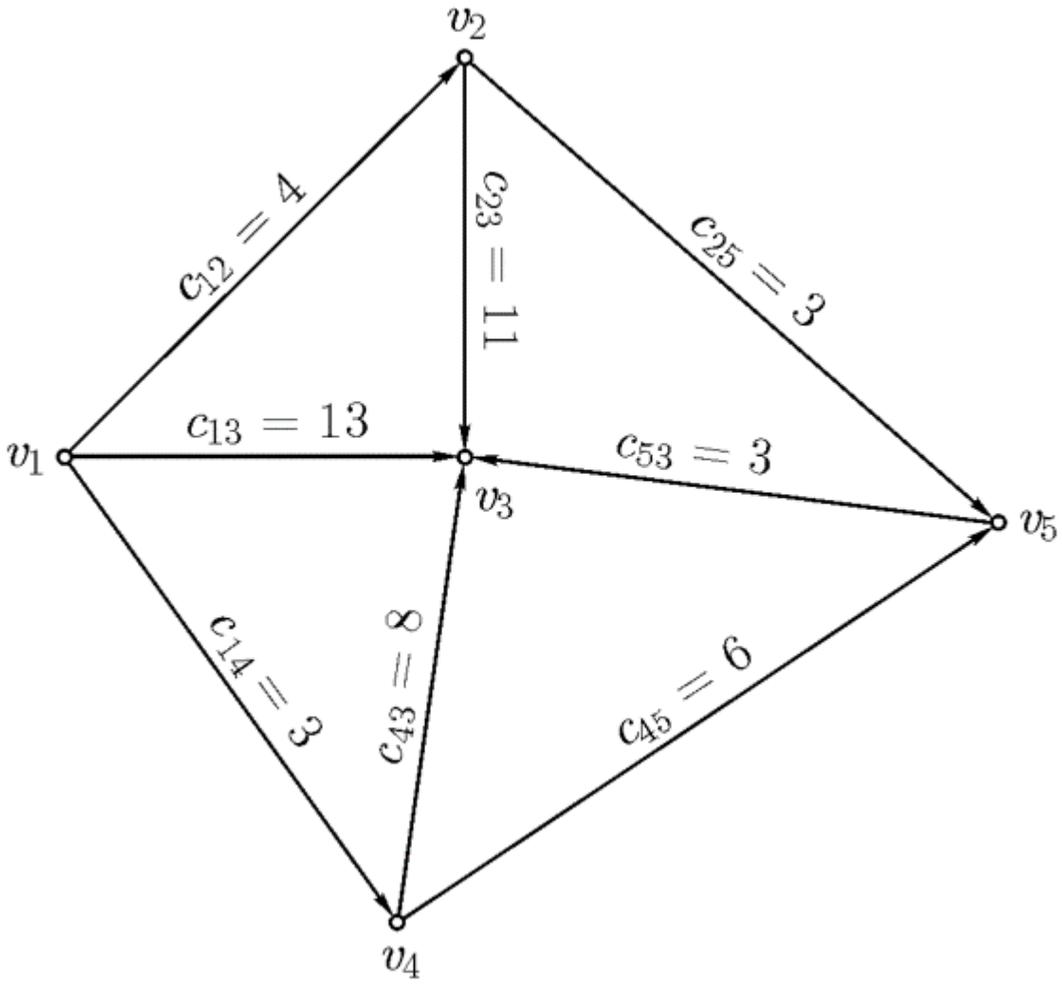
Tutorial example

nodelist_example.txt

1	1
2	4
3	6
4	6
5	8
6	9
7	

arclist_example.txt

1	2	4
2	3	13
3	4	3
4	3	11
5	5	3
6	3	8
7	5	6
8	3	3



Programming in python

General remarks on coding

- Meaningful varibale names

```
p = [None]*n
predList = [None]*n
```

- Comment Your Code

```
labels = [None]*n     #initialize labels
predList = [None]*n     #initialize predecessors
```

- Use functions for repeating tasks

```
def dijkstra(nodelist,arclist,nodepl,...):
    ...
```

Reading in data with NumPy


```
import numpy as np
np.loadtxt()
```

- **fname** Name of the file, e.g. “nodelist.txt”
- **delimiter** String which is used to separate values
- **skiprows** Number of lines to be skipped

see: https://numpy.org/doc/stable/reference/generated/numpy.loadtxt.html

Accessing data in NumPy

- Access a certain column, e.g. the first column:
  - `data1[:, 0]`

- Access a specific value
  - `data1[row][column]`
  - e.g. value in the first row and 6th column: `data1[0][5]`

Example: reading in data with NumPy

```
nodelist = np.loadtxt('nodelist_example.txt',
 skiprows=0, dtype='int')

print(nodelist[4])
    = 8

print(len(nodelist))
    = 6
```

Reading in data with pandas


```
import pandas as pd
pd.read_csv()
```

- **`fname`**: Name of the file, e.g. `"nodelist.txt"`
- **`sep`**: Character used as the delimiter, e.g. `‘\s+’` for whitespaces
- **`names=[“column1", “column2"]`**: Names for columns.

see: pandas.read_csv — pandas 1.5.1 documentation (pydata.org)

Accessing data in pandas

- Access a column
  - **`data1["phi"]`**
- Access a single value:
  - **`data1["phi"][5]`**

Advantage of pandas: columns have names – easy access

- If index should start with 1 and not with 0:
  - **`data1.index = data1.index+1`**

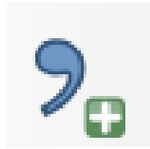
Lists

```
templist=[]
templist.append(2)
templist.append(4)
templist.append(5)

print(templist)
    = [2, 4, 5]
```

```
templist =[2,5,7,8]
templist.remove(7)
print(templist)
    = [2, 5, 8]

del(templist[2])
print(templist)
    = [2, 5]
```

Visualization & presentation

Interface python - QGIS

- QGIS:
  - Import text file as layer:  (Strg+Shift+T)
  - Points to path (Punkte zu Weg) (Strg+Alt+T)

- File nodepl.txt is a CSV-like file (data seperated by spaces)

```
47.0527089     15.4404670
47.0530285     15.4411845
47.0593736     15.4747901
47.0574058     15.4755590
47.0602870     15.4690404
```

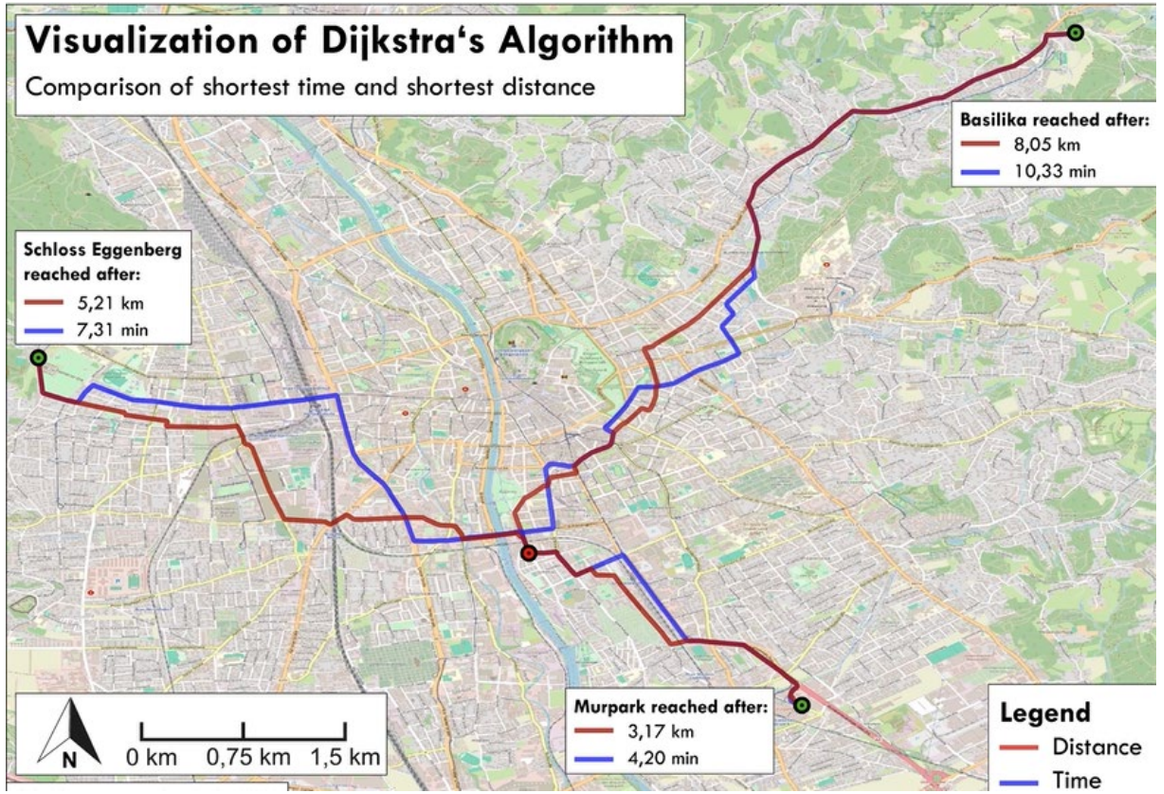
- Create similiar structure for results

```
output='lat\tlon\tnode\n'
for node in route:
    output+="{}\t{}\t{}\n".format(nodepl['lat'][node],nodepl['lon'][node],node)
```

Results and visualization

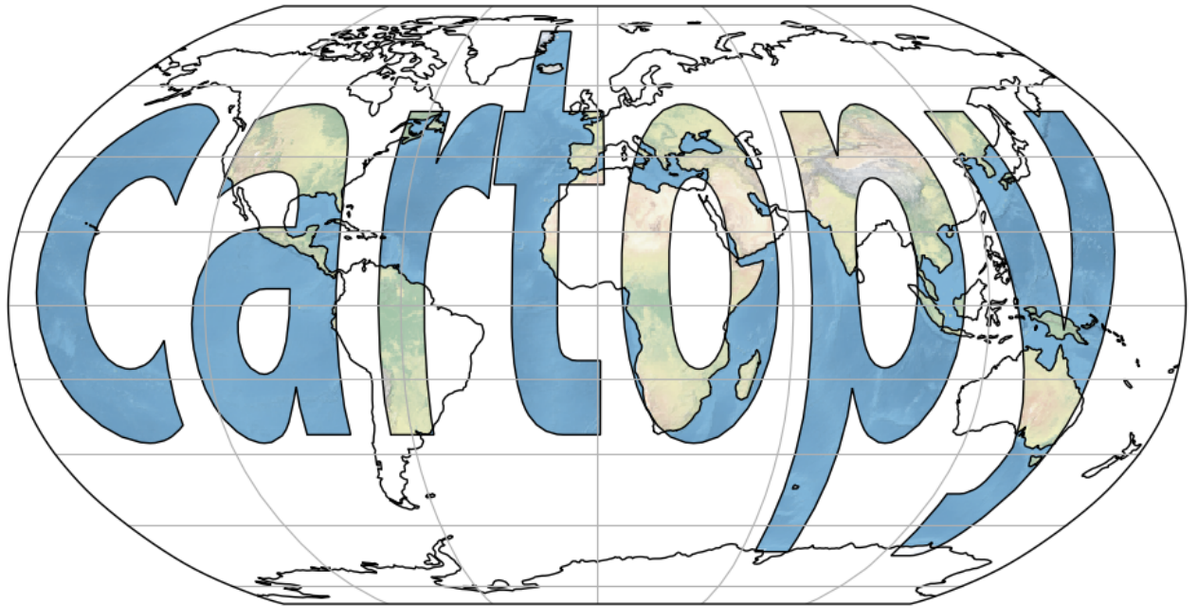
- predList.txt:
  - Store your predecessors
  - Necessary to visualize route

- Visualization: QGIS or cartopy
  - Use appropriate line width
  - Select a useful projection
  - Include all relevant map elements (north arrow, scale bar, etc.)



Visualization with Cartopy

https://cartopy.readthedocs.io/stable/



Create your video

Options

- Record using PowerPoint (Slide Show / Record Slide Show)
- Screen recording
- Film and cut your video

Evaluation

- **Hand in by November 16th, 11:59 p.m. in the TeachCenter**
  - **Video** (*Group#.mp4*), max. 5 Min!
  - **ZIP-Folder**
    - Uploaded by one group member for the whole group
    - Source-Code (*lab01_ws2025_26_Group#_Surname1_Surname2.py**)
    - ReadMe-File („*README.txt*“) containing additional information needed to run your programme (e.g. version of the used programming language, libraries,…)
    - If applicable: additional files needed to run your code

Best presentation award

- 2 best presentations will be rewarded with a small prize and will be put on TUBE