# Multimodal Route Planning

**Claudio Tomasi**, Stefano Gualandi, Davide Duma, Martina Fischetti, Juan Nicolás Ibáñez

Spring Workshop
March 17, 2022

# Contents

# Motivation (1/2)

- European Green Deal
  - Climate neutrality by 2050
- Transport as key player
  - Major user of energy
  - Significant contributor to greenhouse gas emission
- Public transport more sustainable
- Large amount of data
  - Accessibility-to-opportunities measures
  - Time-dependent schedule-based routing problem

## Motivation (2/2)

- Analysis of entire rail network in Europe
  - How many people have access to services by trains
- Routing problem solution to get answers
  - Time-dependent
  - Schedule-based
  - Multimodal
- GOAL: Extension  to all kinds of transport supply
  - Not only trains
  - Complement analysis of public transport performance
  - Computationally challenging

## Multimodal Accessibility Index

- "The ease and convenience of reaching some destination"
- Measure that provides an indicator of accessibility
  - Density of the public transport networks
  - Reachability for a given area
- Depending on network infrastructure
  - Proximity and diversity of the public transport network
  - Frequency of service at accessible nodes

# Literature Review


- Dijkstra, Edsger W., "A note on two problems in connexion with graphs" *Numerische mathematik* 1.1 (1959)
- Delling, Daniel, et al. "Round-based public transit routing" *Transportation Science* 49.3 (2015)
- Witt, Sascha. "Trip-based public transit routing" *Algorithms-ESA* (2015)
- Dibbelt, Julian, et al. "Connection scan algorithm" *Journal of Experimental Algorithmics (JEA)* 23 (2018)
- Baum, Moritz, et al. "UnLimited TRAnsfers for Multi-Modal Route Planning: An Efficient Solution" *arXiv preprint arXiv:1906.04832* (2019)

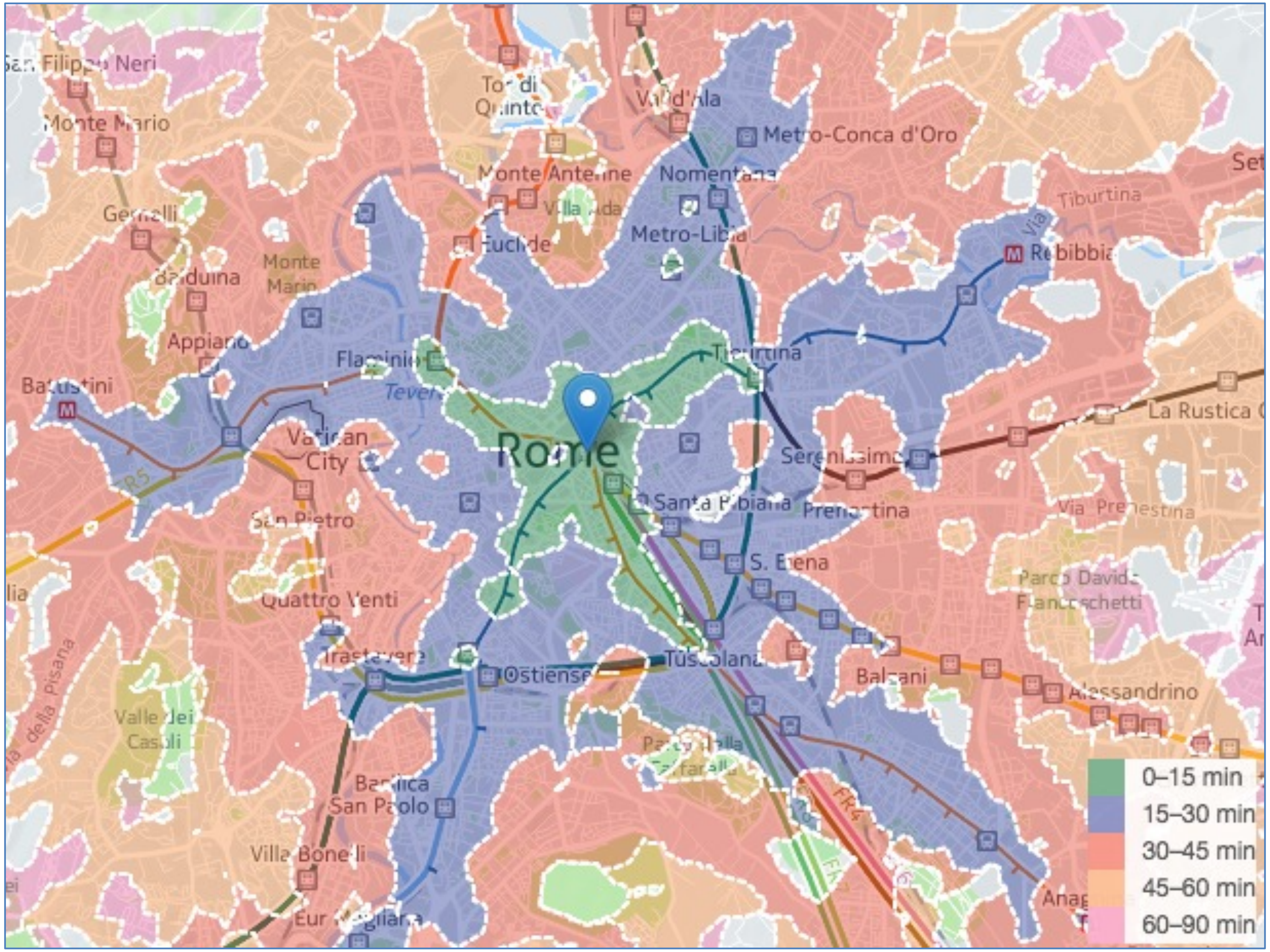
## Graphs

- Mathematical structures modelling pairwise relations between objects
- $G = (V, E)$
  - $V$ set of vertices (e.g., bus stops)
  - $E$ set of edges, i.e., $e = (v, w)$, with associated length
- Path as sequence of edges joining a sequence of vertices
  - Length of a path is the sum of its edge length

# Shortest Path Problem

- Given a directed graph, find path with minimum costs connecting points
- Depending on what we need, we employ different query types
  - one-to-one for simple source-to-destination
  - one-to-all for isochrone maps
  - all-to-one for choosing the best starting point
  - all-to-all for computing a reachability matrix

# Isochrone Maps

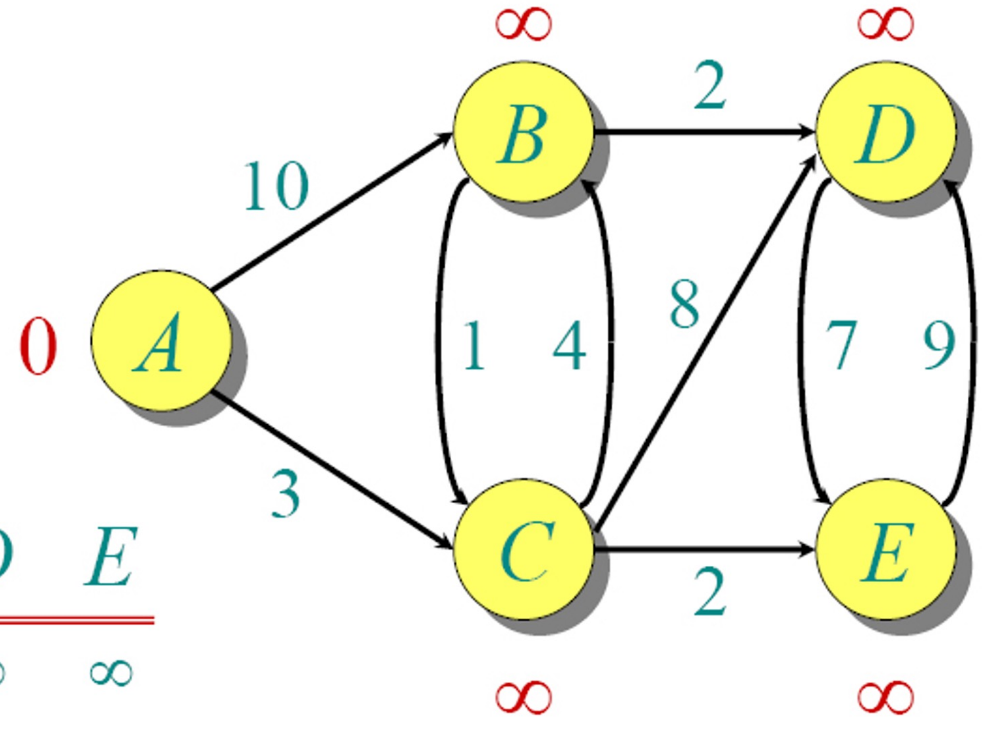


# Dijkstra's Algorithm

- Algorithm to find shortest paths[1]
  - e.g., in road networks
- one-to-all type of query
  - Source in input
  - At the end, a shortest path cost associated to each node
- Data structure for storing and querying partial solutions sorted by distance from the start
  - $O(|V|^2 + |E|)$ using an array
  - $O((|E| + |V|) \log |V|)$ using an adjacency list

(1) Dijkstra, Edsger W., "A note on two problems in connexion with graphs." (1959)

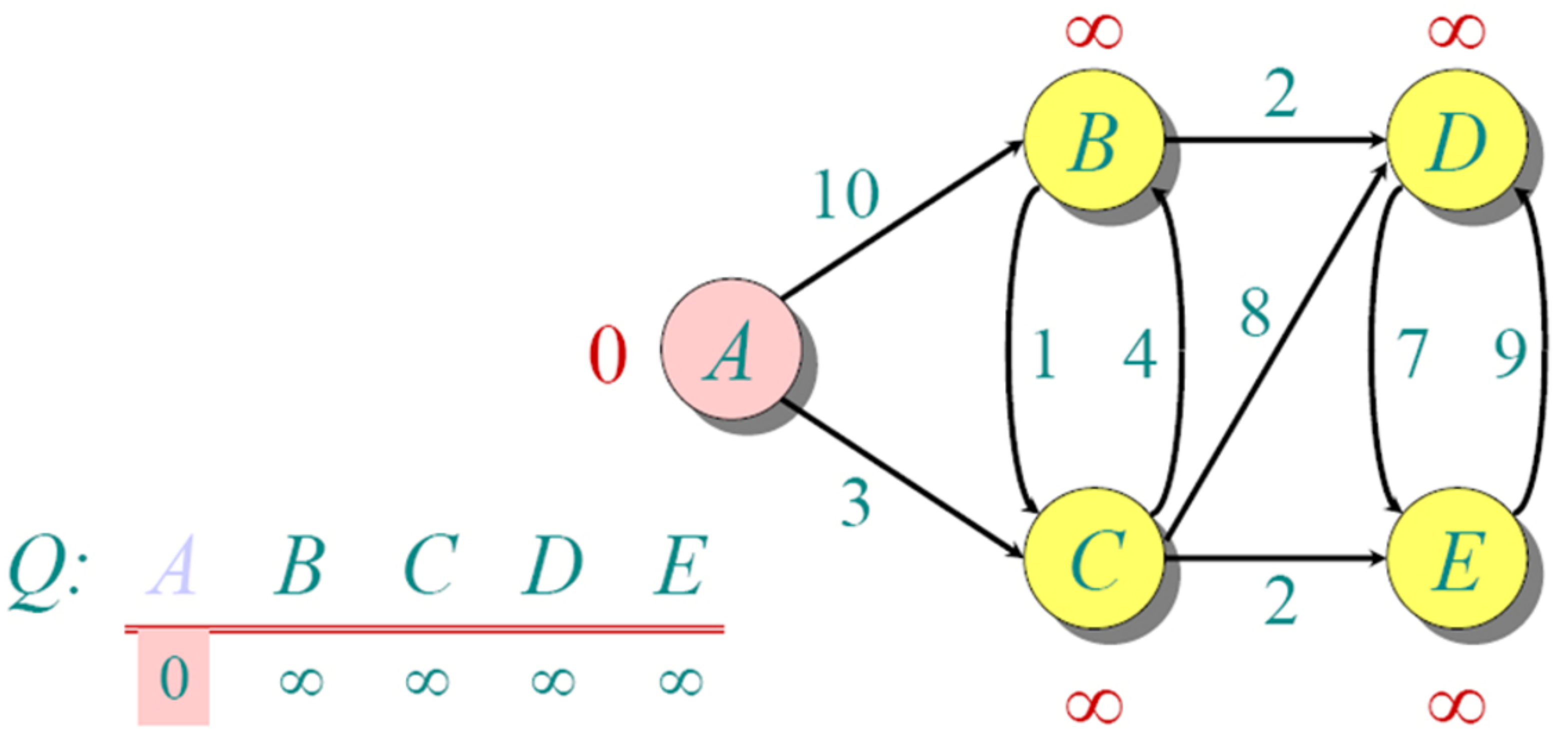
# Dijkstra's Algorithm



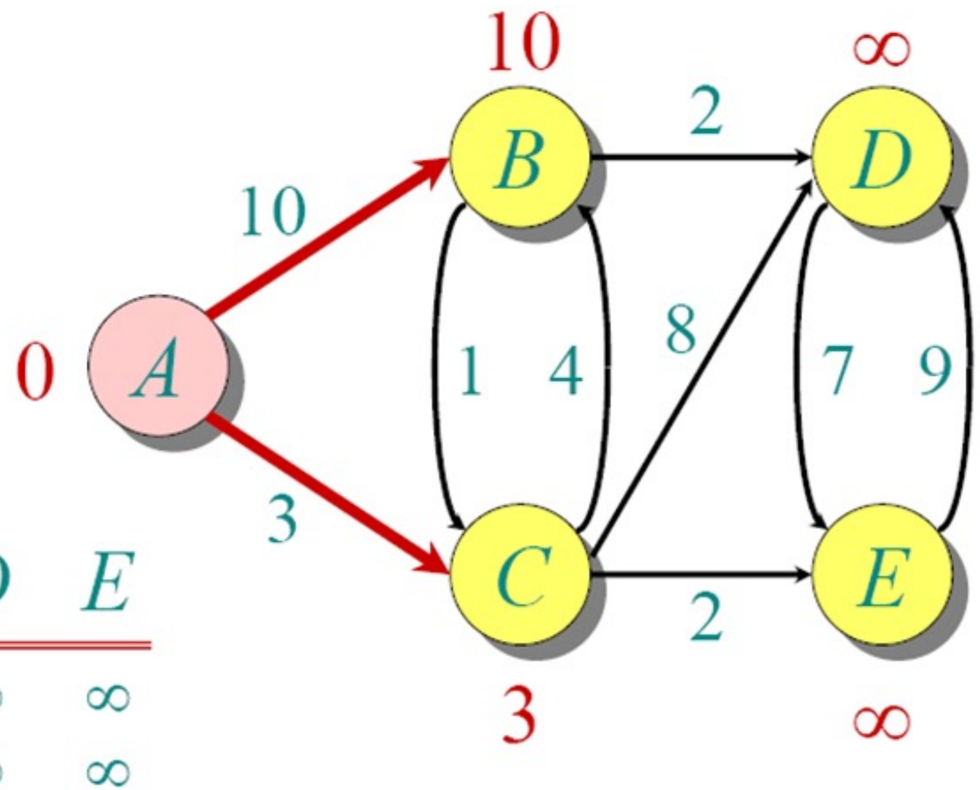

| Q: | A | B | C | D | E |
|---|---|---|---|---|---|
| | 0 | ∞ | ∞ | ∞ | ∞ |

S: {}

# Dijkstra's Algorithm



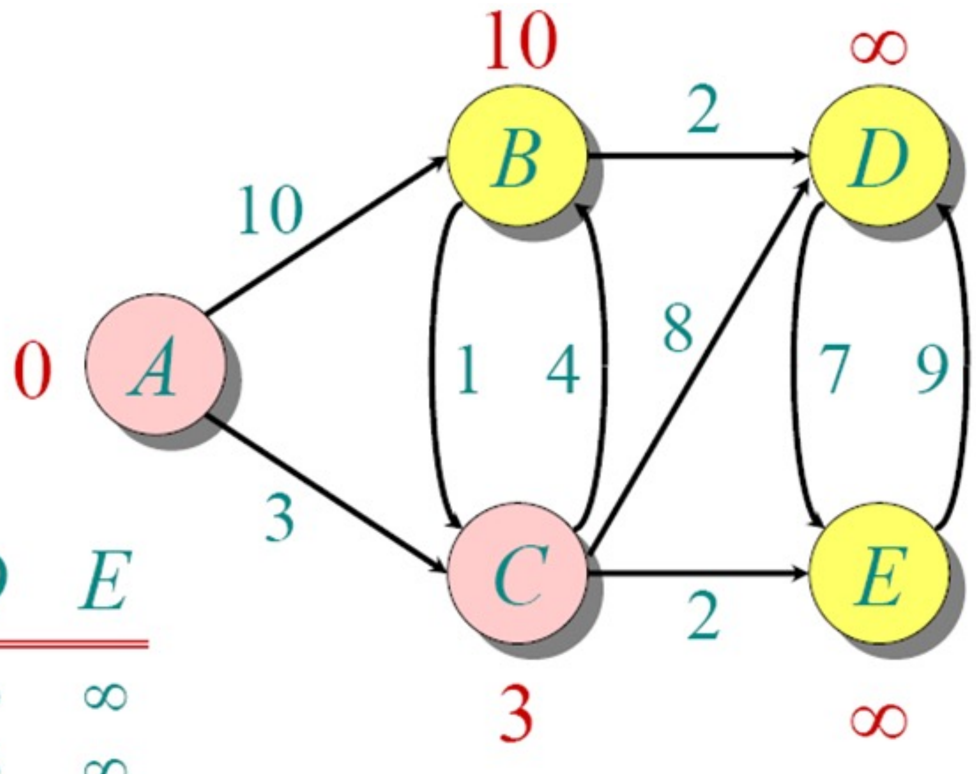
# Dijkstra’s Algorithm




| Q: | A | B | C | D | E |
|---|---|---|---|---|---|
| | 0 | ∞ | ∞ | ∞ | ∞ |
| | | 10 | 3 | ∞ | ∞ |

S: { A }

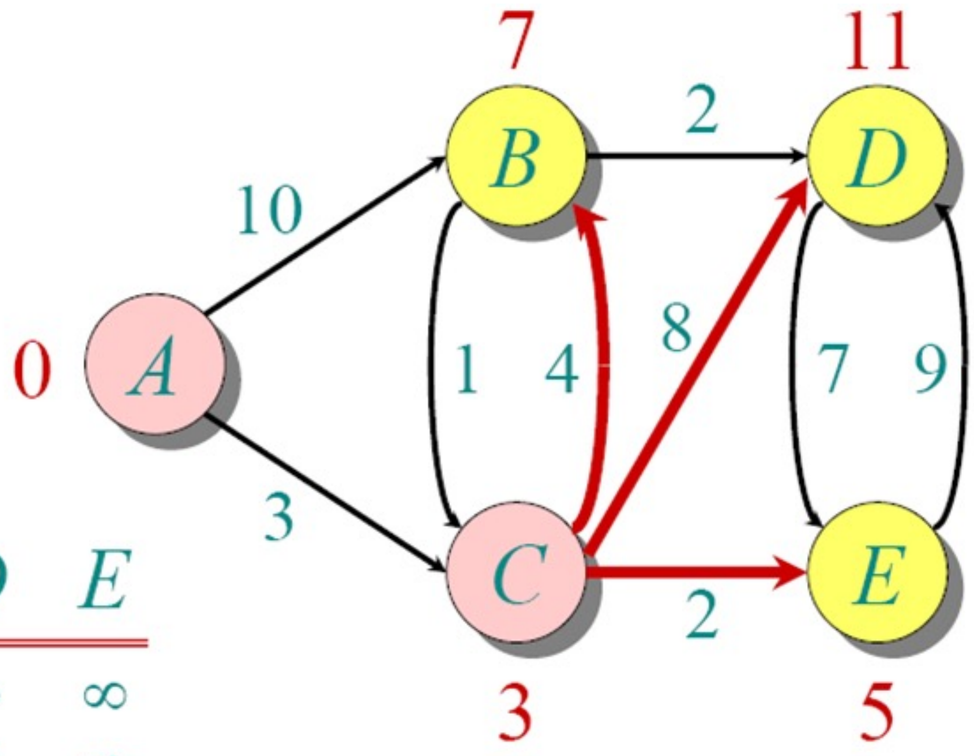
# Dijkstra's Algorithm




| Q: | A | B | C | D | E |
|---|---|---|---|---|---|
| | 0 | ∞ | ∞ | ∞ | ∞ |
| | | 10 | 3 | ∞ | ∞ |

S: { A, C }

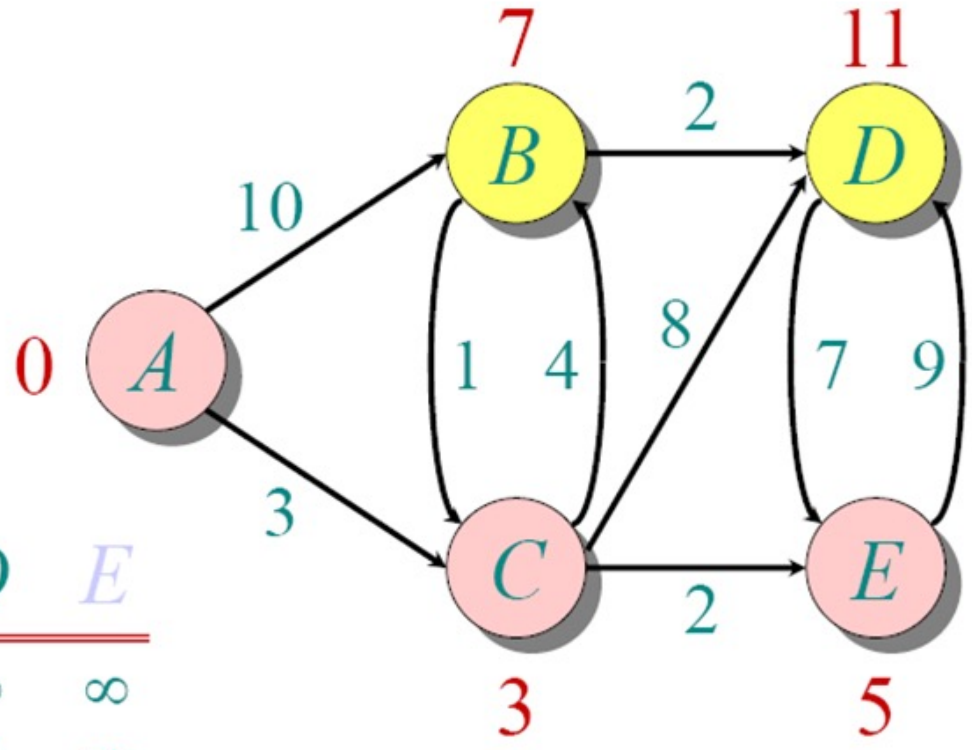
# Dijkstra's Algorithm




| Q: | A | B | C | D | E |
|---|---|---|---|---|---|
| | 0 | ∞ | ∞ | ∞ | ∞ |
| | | 10 | 3 | ∞ | ∞ |
| | | 7 | | 11 | 5 |

S: { A, C }

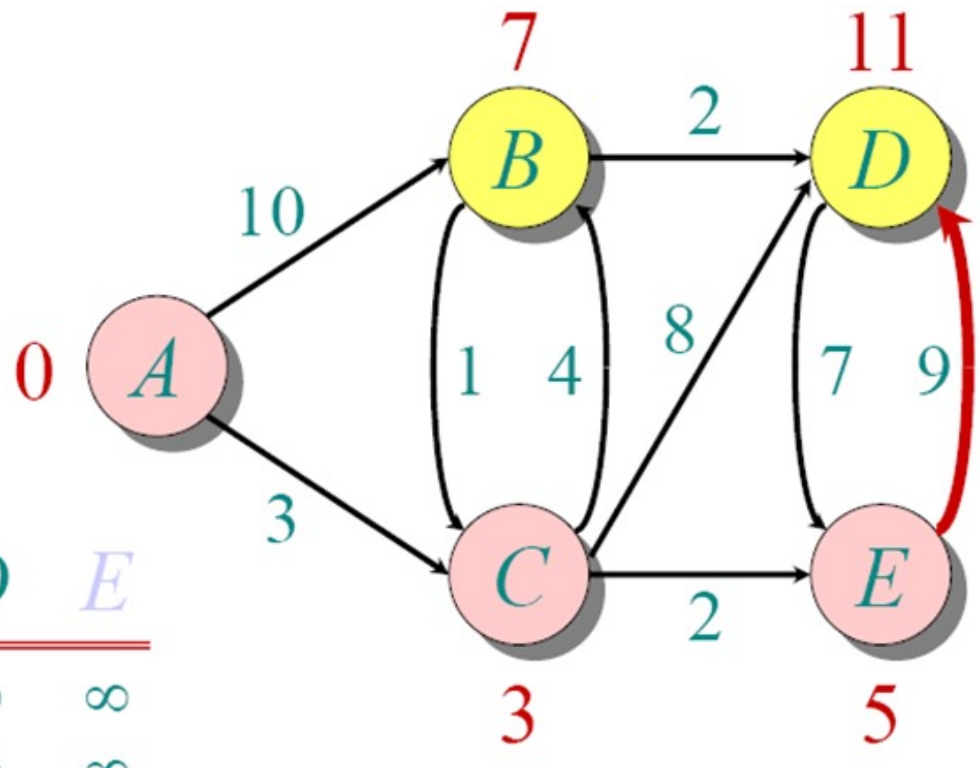
# Dijkstra’s Algorithm




| Q: | A | B | C | D | E |
|---|---|---|---|---|---|
| | 0 | ∞ | ∞ | ∞ | ∞ |
| | | 10 | 3 | ∞ | ∞ |
| | | 7 | | 11 | 5 |

S: { A, C, E }

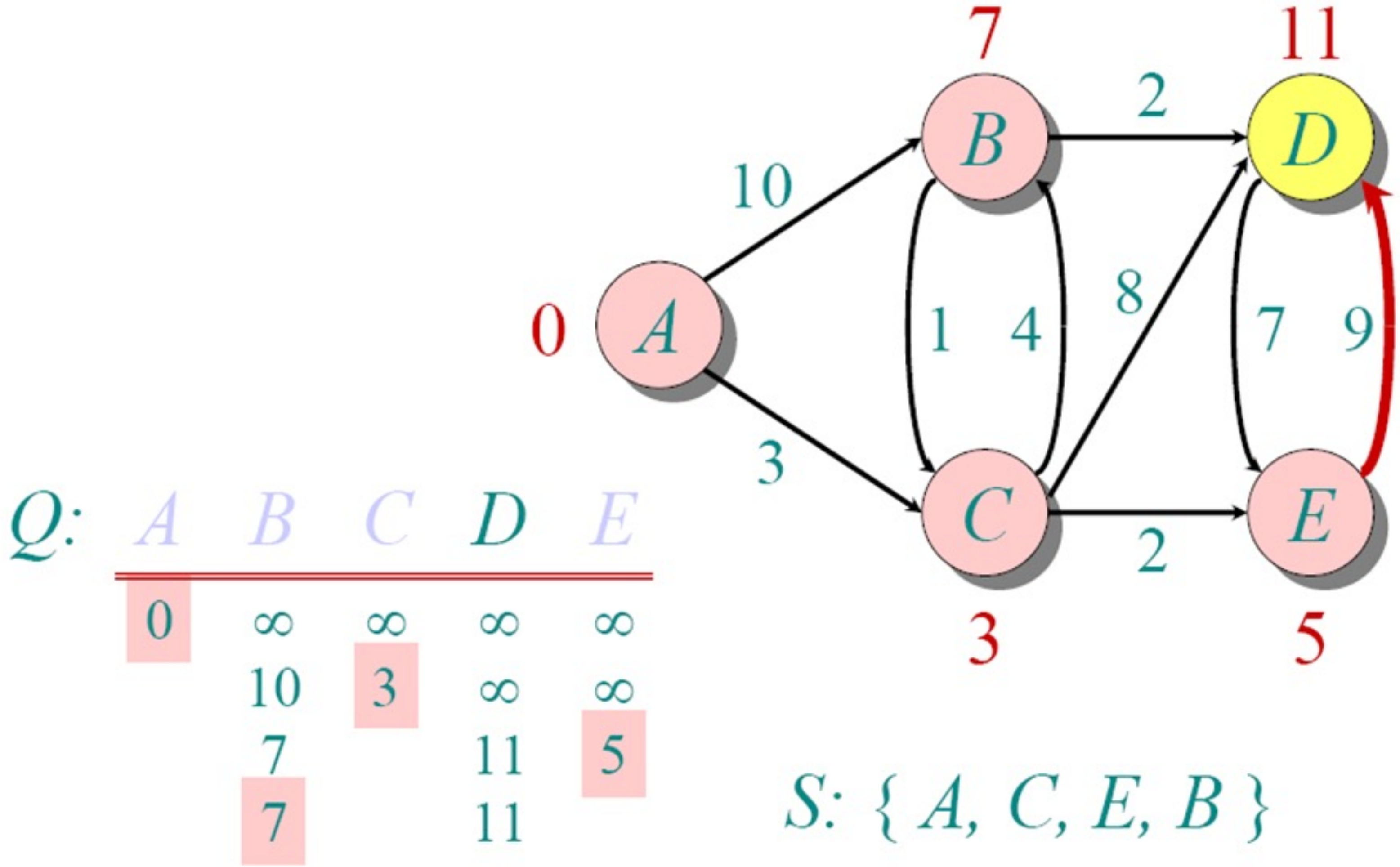
# Dijkstra's Algorithm



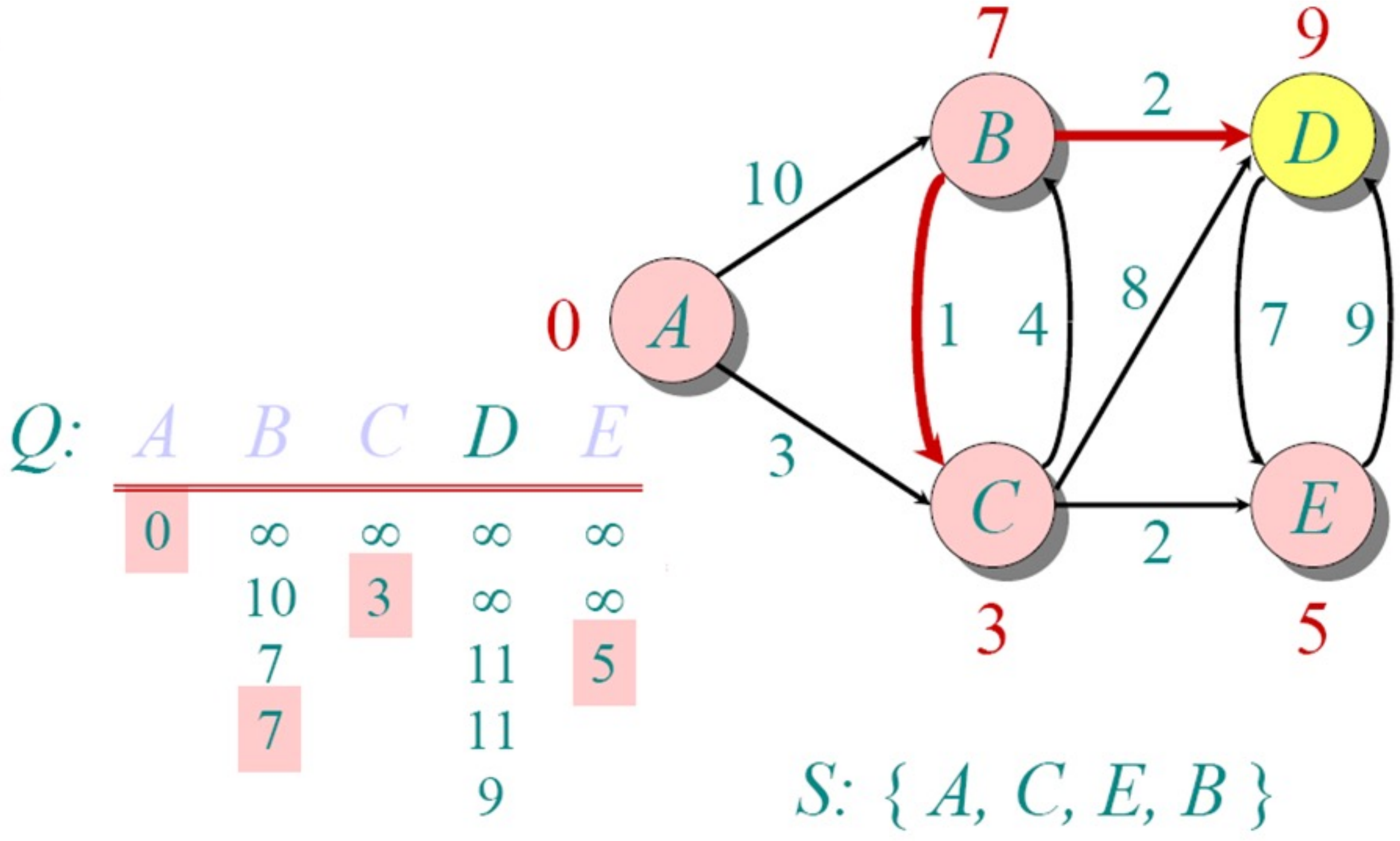

| Q: | A | B | C | D | E |
|---|---|---|---|---|---|
| | 0 | ∞ | ∞ | ∞ | ∞ |
| | | 10 | 3 | ∞ | ∞ |
| | | 7 | | 11 | 5 |
| | | 7 | | 11 | |

S: { A, C, E }

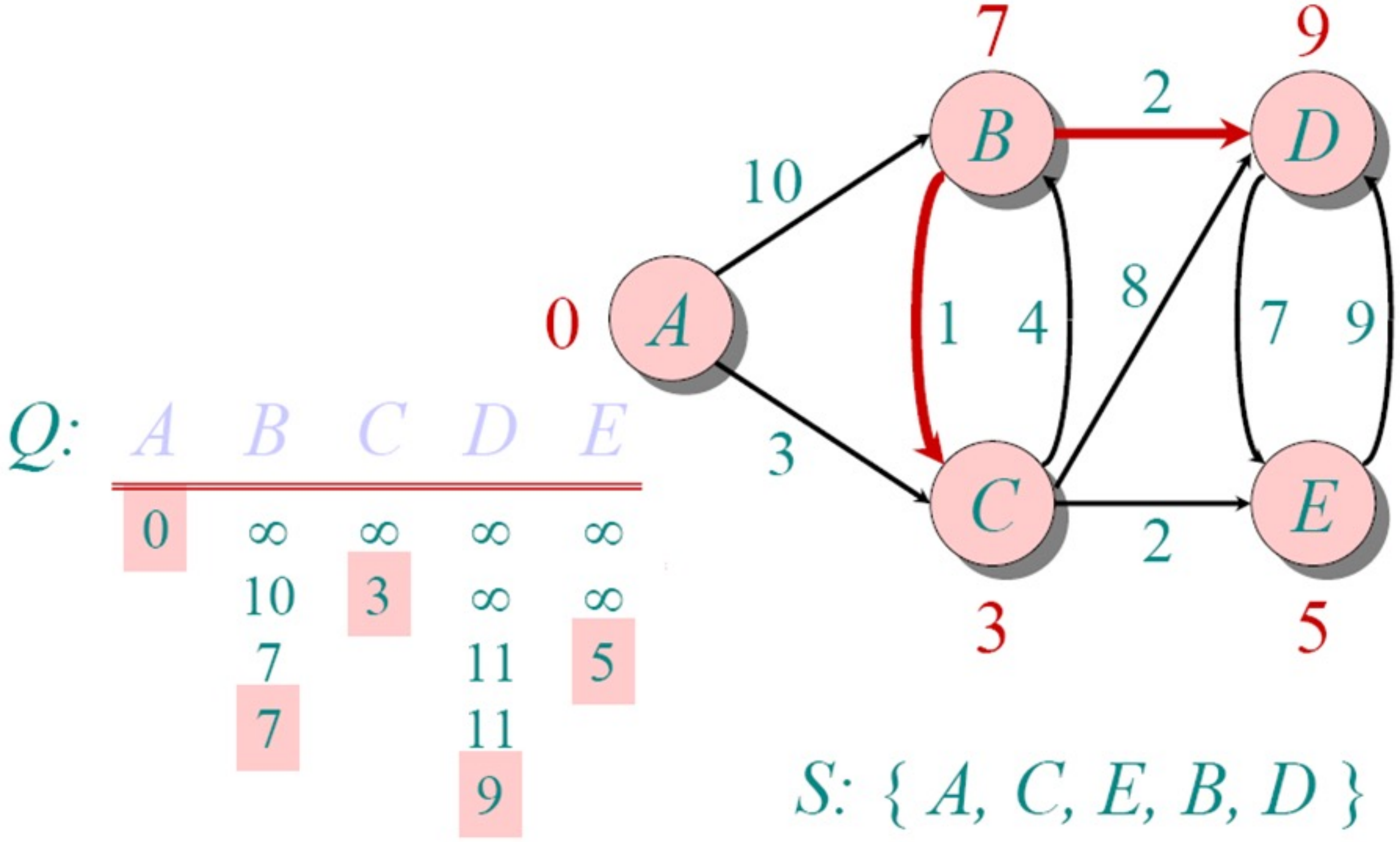
# Dijkstra's Algorithm



# Dijkstra's Algorithm



# Dijkstra’s Algorithm



## Schedule Based Networks

- Network structure time-dependent
- Public transport timetable
  - Set of stops
  - Set of routes
  - Set of trips
- Accessibility only at specific points in time
- Appropriate modeling to enable a fast computation of journeys
  - Sequences of trips a traveler can take.
  - Not only simple paths minimizing the cost, but several optimization criteria to get the desired solution

## Multimodal Networks

- Integrated transportation network
  - Unrestricted networks (road, etc..)
  - Schedule-based public transport service
- Overlapping of graph vertices and stops
- Permit multimodal query
  - Considering a preference on the vehicle
  - Optimizing the transfers between routes
  - Graph algorithm not the best choice anymore

# RAPTOR Algorithm

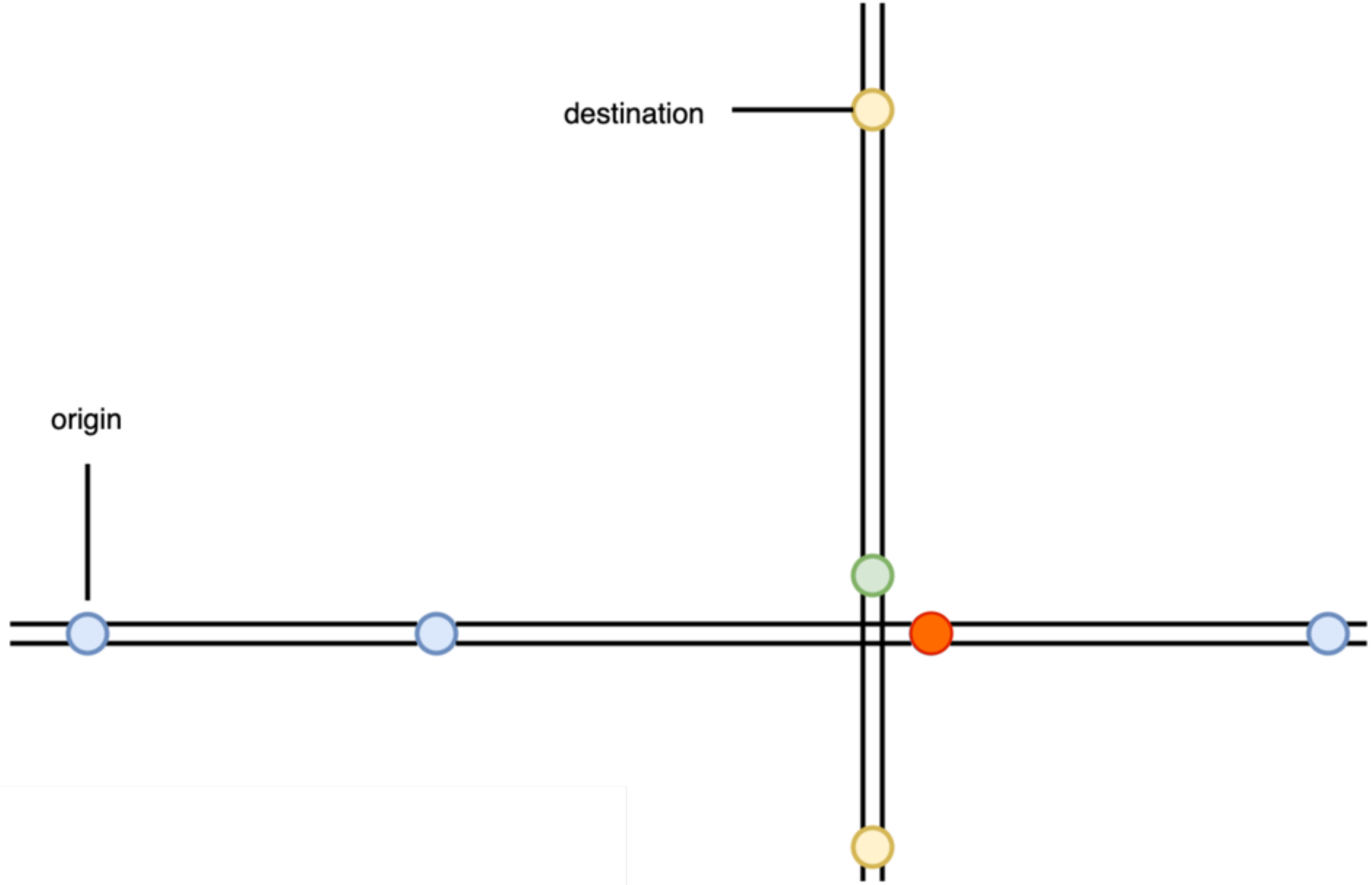
- Round-Based Public Transit Routing[2]
  - Non-Dijkstra approach
  - Directly exploitation of timetable, without graph modelling
  - Works in rounds (iterations)
- Standard RAPTOR
  - For two given stops, computes optimal journeys, minimizing the arrival time number of transfers
- Extensions
  - McRAPTOR for several criteria
  - rRAPTOR for range queries (departure time window)

(2) Delling, Daniel, et al. "Round-based public transit routing." Transportation (2015)

## RAPTOR Algorithm

- For each iteration, we have three phases
  - For all available stops, it finds the trips associated to them that can be boarded at a given time
  - Find all the stops reachable using those trips and compute the time needed to reach them, updating their labels
  - At each round, add a new transfer, i.e., footpaths between nearby stops, in the total number of transfers allowed

# RAPTOR Algorithm



# Numerical Results

| | London | Switzerland | Germany |
|---|---|---|---|
| **Stops** | 19’682 | 25’125 | 243’167 |
| **Routes** | 1’995 | 13’786 | 230’225 |
| **Trips** | 114’508 | 350’006 | 2’381’394 |
| **Stop Events** | 4’508’644 | 4’686’865 | 48’380’936 |

- RAPTOR query performance
  - London 5.2 ms
  - Switzerland 12.3 ms
  - Germany 344.5 ms

## What Next?

- EU-wide queries
  - Tackle the challenge of multimodal routing problems
  - all-to-all queries for accessibility information
  - Smart implementation for fast computation
- Results
  - Understanding how to define transport policy
  - Add/remove/modify connections

# Thank you for your attention