COURSE INTRODUCTION LECTURE 1

CMPE 252, Fall 2023

Artificial Intelligence and Data Engineering

Jun Liu

Special Accommodations

- Any student who feels that he or she may need an accommodation because of a disability (learning disability, attention deficit disorder, psychological, physical, etc.), please make an appointment to see me during office hours.

ABOUT ME

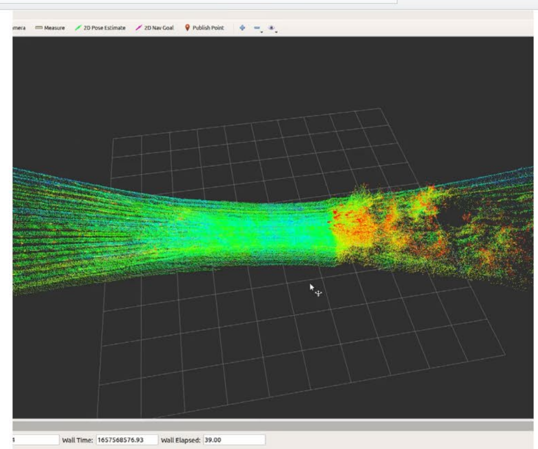
- PhD in Computer Engineering from Virginia Tech, 2022
- New Assistant Professor in CMPE
- Research interests:
  - – Distributed multi-agent systems
  - – Distributed decision-making, optimization, and estimation
  - – Heterogeneous multi-robot coordination
  - – Large-scale systems, such as the Internet of Things (IoT) and swarms

ABOUT ME

- j.liu@sjsu.edu
- ENG 261
- Office hours: Thursday, 4-6 PM, in-person or by appointment

ABOUT ME


00:00.00


Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Wall Time: 1657568576.93
Wall Elapsed: 39.00
Zoom. Shift: More options.

ABOUT ME

- Research topics:
  - Communication, e.g., reliability
  - Perception, e.g., noise
  - Computing, e.g., load balance
  - Safety, e.g., attack, failures
  - Autonomy, e.g., decision-making
  - Modeling, e.g., accuracy
  - etc.

RESEARCH OPPORTUNITY

- If you want to do research with me:
  - Talk with me @ office hours or after class
  - Schedule a Zoom meeting
  - Email
  - Etc.

COURSE INTRODUCTION

- So, what's this course about?

- We will focus on the fundamental problems in AI
  - ❑Planning, e.g., autonomous driving path planning
  - ❑estimation, e.g., Kalman Filtering, IoT
  - ❑search, e.g., shortest path
  - ❑Programming, e.g., dynamic/linear/integer programming
  - ❑DL, RL, e.g., ChatGPT
  - ❑Graph, e.g., data structure
  - ❑Etc.

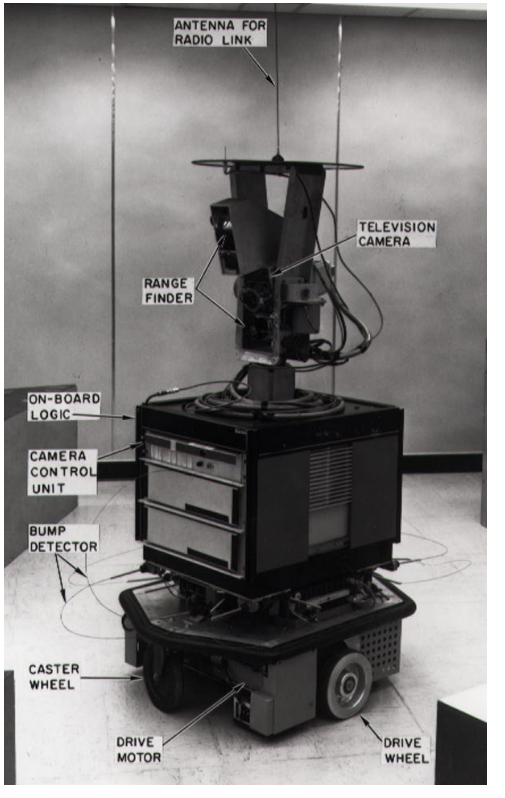
Example: path planning – autonomous driving

- How should the robot go from point A to point B?

- We will start simple:
    - Point robot that can move in any direction
    - Known environment with stationary obstacles
    - Perfect sensing
    - Perfect control

Example: path planning – autonomous driving

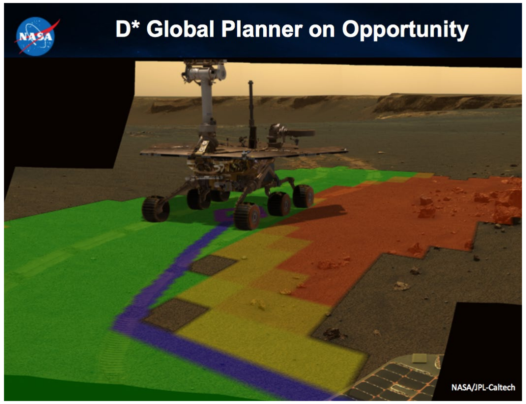
- Invented by Hart, Nilsson and Raphael of Stanford Research Institute in 1968 for the Shakey robot



Example: path planning – autonomous driving

- How should the robot go from point A to point B?

- We will start simple:
  - Point robot that can move in any direction
  - Unknown environments and/or moving obstacles
  - Perfect sensing
  - Perfect control

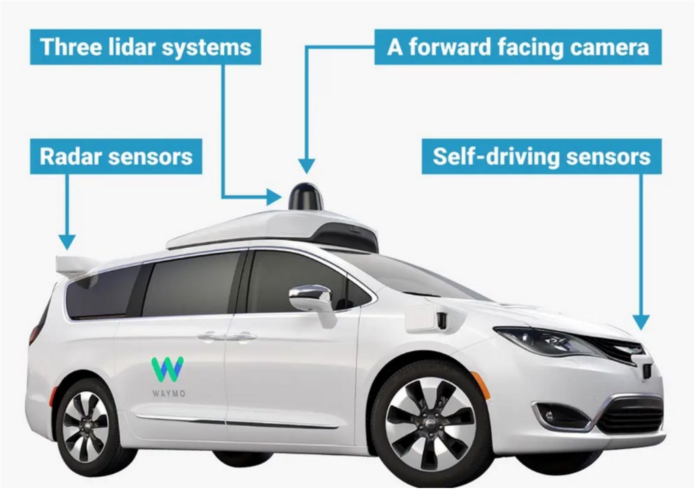


- Challenges?

Example: path planning – autonomous driving

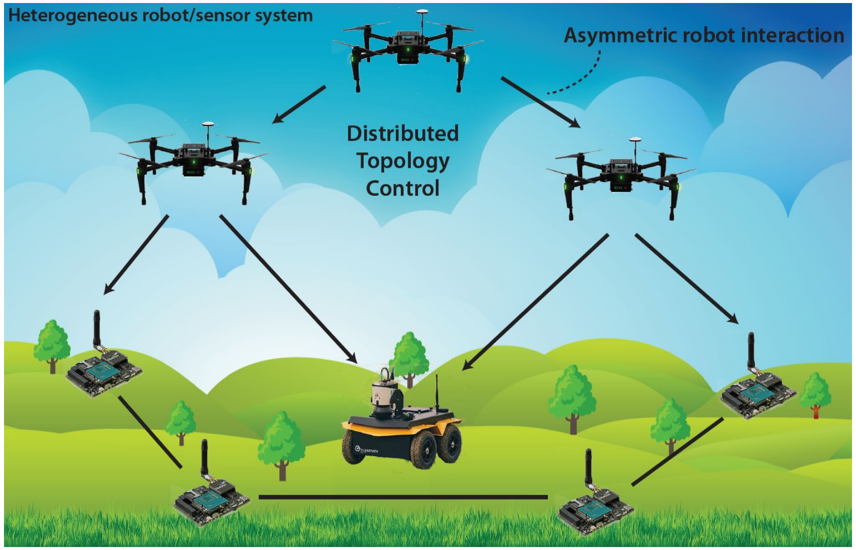
- How should the robot go from point A to point B?

- We will start simple:
  - Point robot that can move in any direction
  - Complex, high dimensional environments
  - Perfect sensing
  - Perfect control



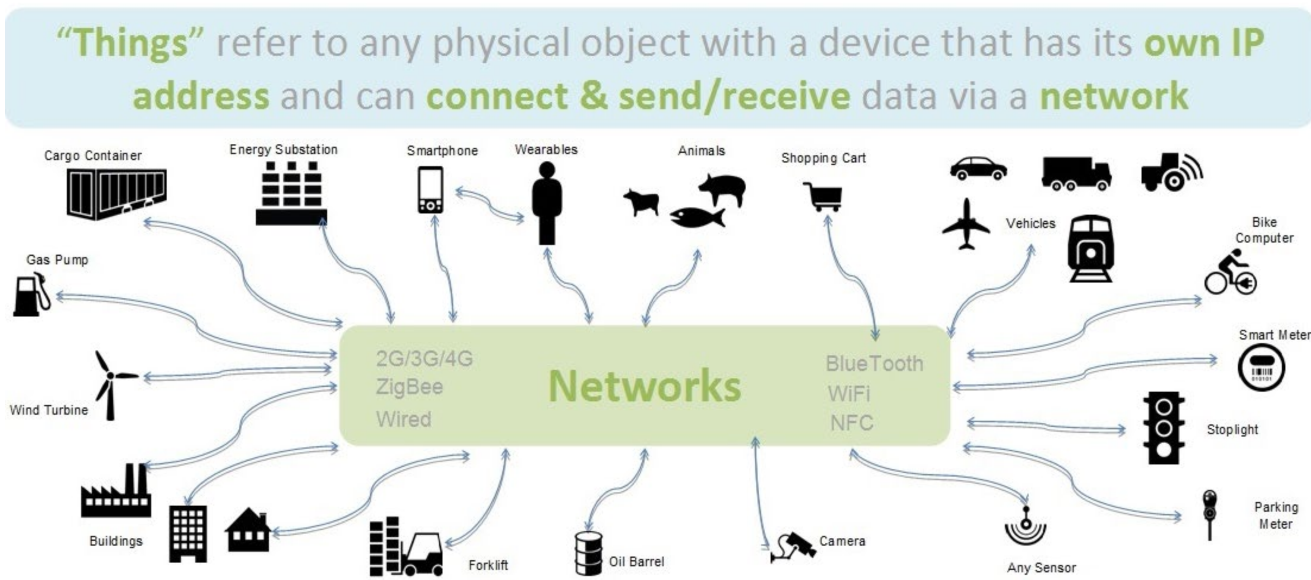
- Challenges?

Example: heterogeneous multi-robot systems

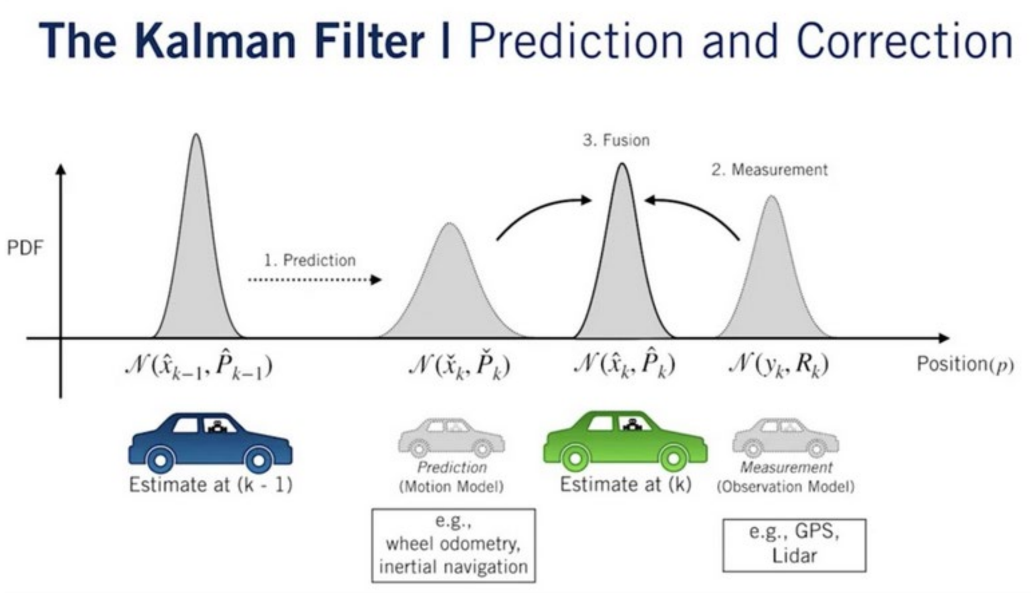


What are the challenges?

Example: Wireless / sensor networks, IoT



Example: Kalman filtering



Prerequisites

- Comfortable with data structures
    - graph, tree, queue, linked list, etc.
- Basics of probability
    - typical distributions, pdf, mean and covariance of random variables
- Comfortable programming in at least one of MATLAB, Python, C/C++ and Linux
    - will be implementing many of the algorithms studied in class
    - there will be NO coding support

CLASS FORMAT

- Homework:
  - You usually have two weeks
- Paper presentation:
  - Select one paper for paper presentation, Paper list will be released on 8/28, due by 9/5
  - No implementation, but there are some requirements
- Midterm:
  - In-class. closed books/handouts/laptops/calculators.
- Final Project:
  - Select another paper for your project
  - Proposal (w/ short presentation) + presentation + implementation + report

** The format is subject to change depending of enrollment numbers*

PAPER SELECTION

- A google document will be used for paper selection.
- Selections will be first-come, first-served. Only 1 student per paper!
- Paper selection: will be released on 8/28 &due by 9/5.
- Everyone is expected to select two papers: 1 presentation + 1 final project
  - One paper will be used for your in-class presentation
  - One paper of your choosing will be extended in a project.

*If anyone is unhappy with their selections for the final project or would like to suggest paper(s), please email me.

ACCESSING PAPERS

- Paper list will be posted on class website (more on this later).

- All papers should be accessible through standard sources, e.g.:
  - Google Scholar
  - IEEE Xplore
  - arXiv

- For details on accessing research publications go to library.sjsu.edu.

PAPER & PROJECT PRESENTATION

- Presentations should convince the audience that you wrote the paper.
- We are not interested in a copy of the paper!
- Tell us:
  - why the paper is important
  - how it innovated
  - how you think it could be applied or extended.
- Again, presentations must be 15-20 minutes in length.
- Everyone in class will be expected to have read the abstract and introduction of the paper.
- Participation in the discussion time after each presentation will be expected throughout the year.

PROJECT

- Each student will choose a paper to extend
- Everyone need to submit a project proposal by 11/6.
  - 1-2 pages, IEEE conference format, word or latex
  - Please describe you want to do, what are the challenges, why are they important
  - Important: include what is your planned extension part

PROJECT

- The project report must be submitted as a paper of 6 pages (standard IEEE conference format) describing the project outcome.
  - It is fine if you use some part of your proposal.
- These final project presentations will focus more on the extension the student has made, than the original paper.
- If instead a new idea was proposed, the presentation will describe the novel contribution.
- The project should ideally be the core of a potential conference paper submission. Students can come to me to discuss ideas and to brainstorm.
- Final project report due by 12/5 (the last class meeting)

PROJECT PROPOSAL PRESENTATION

- Before the final project, everyone will give short/quick proposal presentation.
- This presentation is used as a guideline to pace your final project.
- Presentations should focus on what you will do
- Presentations can be 5-10 minutes.
- Everyone in class will be expected to have read the abstract and introduction of the paper if the topic is from the paper list.
- Participation in the discussion time after each presentation will be expected throughout the year.

PROJECT IMPLEMENTATION

- Your final project must be accompanied by an implementation.
- Use the framework of your choice, e.g., MATLAB or Python.
- Implement a simulation of an important aspect of the paper.
- Implementation will allow for visuals in the presentation and a deeper understanding of the material.
- It will also prove to us that you have digested the paper completely.
- Some papers are complex and a complete reproduction of the paper in simulation is not necessary.
- The core ideas should be the focus; do not try to prove to us that the method works (it should, the paper is published), instead prove that you understand the core ideas deeply.
- Implementations & report need to be submitted.

GRADING *

- Homework: 30%
  - About 5 assignments
- Midterm: 25%
- Paper presentation: 15%
  - No implementation, but there are some requirements
  - Please remember selection due date
- Final Project: 30%
  - Proposal (w/ quick presentation) + project presentation + implement (10%) + report (10%):

** subject to change depending of enrollment numbers*

GRADING POLICY

- If you submit by due date – no penalty
- Up to one hour late – 5% penalty
- Up to 24 hours late – 20% penalty
- Beyond 24 hours – 100% penalty

- All submissions due by 9 pm on the due date
- This ensures you will have a good sleep

LOGISTICS

- The course material will be on Canvas
- Make sure to check canvas & the online schedule regularly for updates
- Ensure that you are signed up for notifications

SUMMARY

- Your duty:
  - 5 homework
  - 1 short/quick proposal presentation
  - 1 paper presentation
  - 1 project presentation + implementation + report

IMPORTANT DATES

- Paper list will be released on 8/28.
- Paper selection due by 9/5.
- Project proposal due by 11/6.
- Project proposal presentation will begin 11/7.
- Paper presentation will begin 11/14.
- Project presentation will begin 11/28.
- Project report due by 12/5.
- Midterm: 10/17

- All dates will be posted on Canvas.

SURVEY: CLASS BREAK TYPE

- One 15m break

  vs.

- two 10m breaks.

- Which on do you prefer?

Welcome

Let’s get started!

CMPE252
Artificial Intelligence and Data Engineering

Dijkstra's algorithm

Path planning definition & properties

Definition:

- Finding a **continuous** path connecting a system from an initial to a final goal.

Required features:

- Safety
- Collision –free
- Efficience, time & distance
- etc.

Applications:

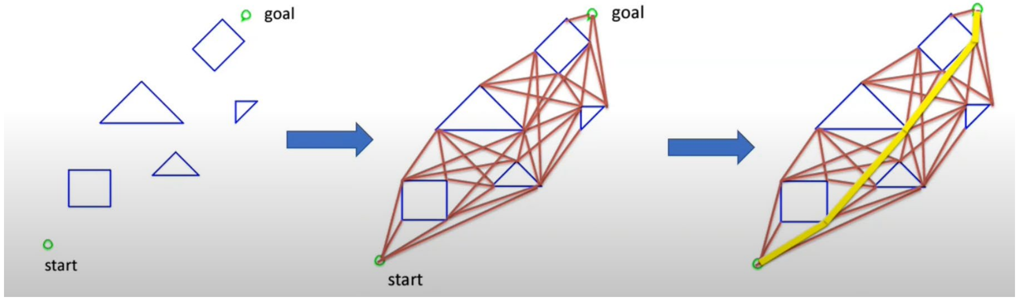
- Warehouse applications, manufacturing, safety and patrolling, auto-driving, map navigation, etc.

Visibility graph

- Assume: robot is a point in 2D, obstacles are 2D polygons.
- Construct visibility graph:

  Nodes:
  - Starting point, goal points, vertices of obstacles

  Edges:
  - Connect all nodes which are visible
  - Include all edges of polynomial obstacles.
- For graph based algorithms, e.g., Dijkstra, A*
- Implement any graph search algorithm, e.g., Dijkstra, from start to goal.



Dijkstra’s algorithm

Dijkstra's algorithm –

- A solution to the single-source shortest path problem in graph theory.
- Works on both directed and undirected graphs.
- Optimal.

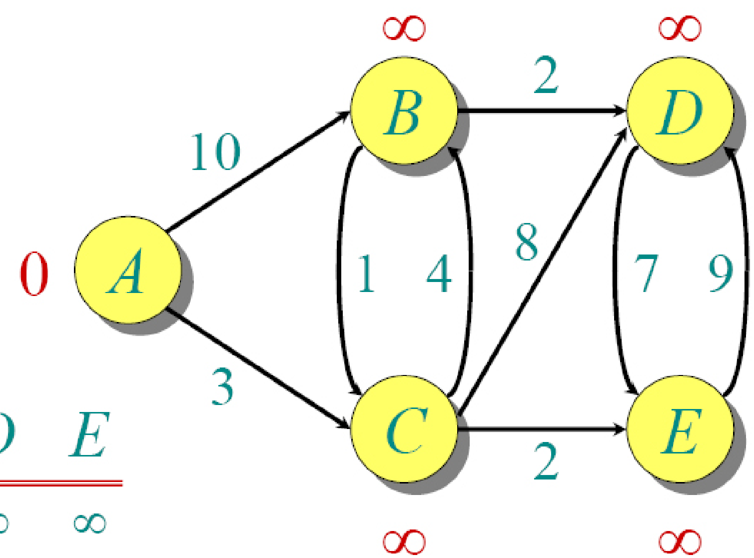
Approach: Greedy

Input: Weighted graph G={E,V} and source vertex v∈V, such that all edge weights are nonnegative

Output: Lengths of shortest paths (or the shortest paths themselves) from a given source vertex v∈V to all other vertices

Dijkstra animated example

Initialize:

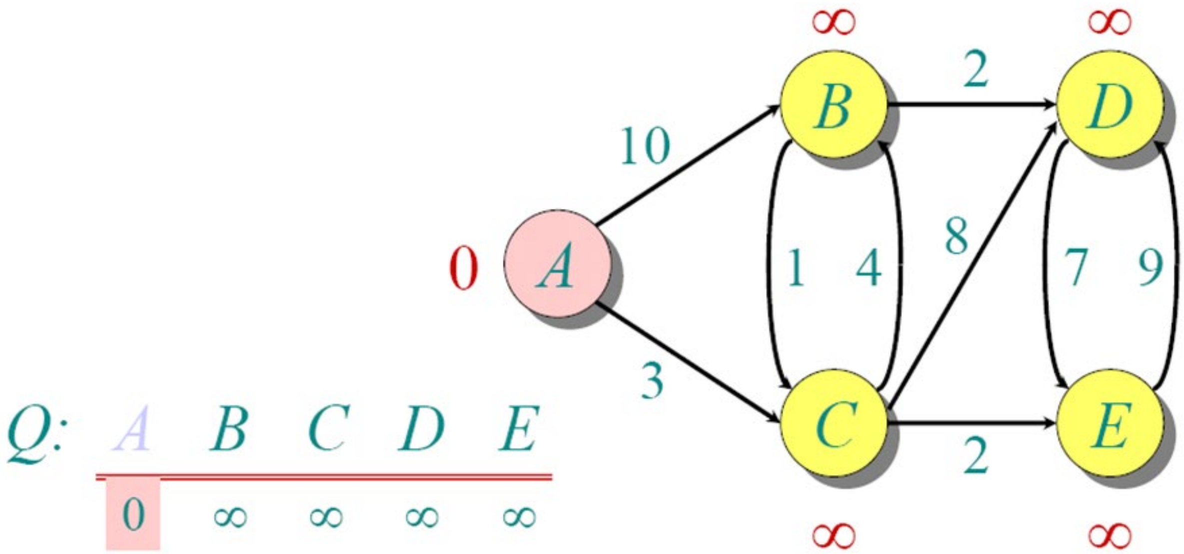


Q:

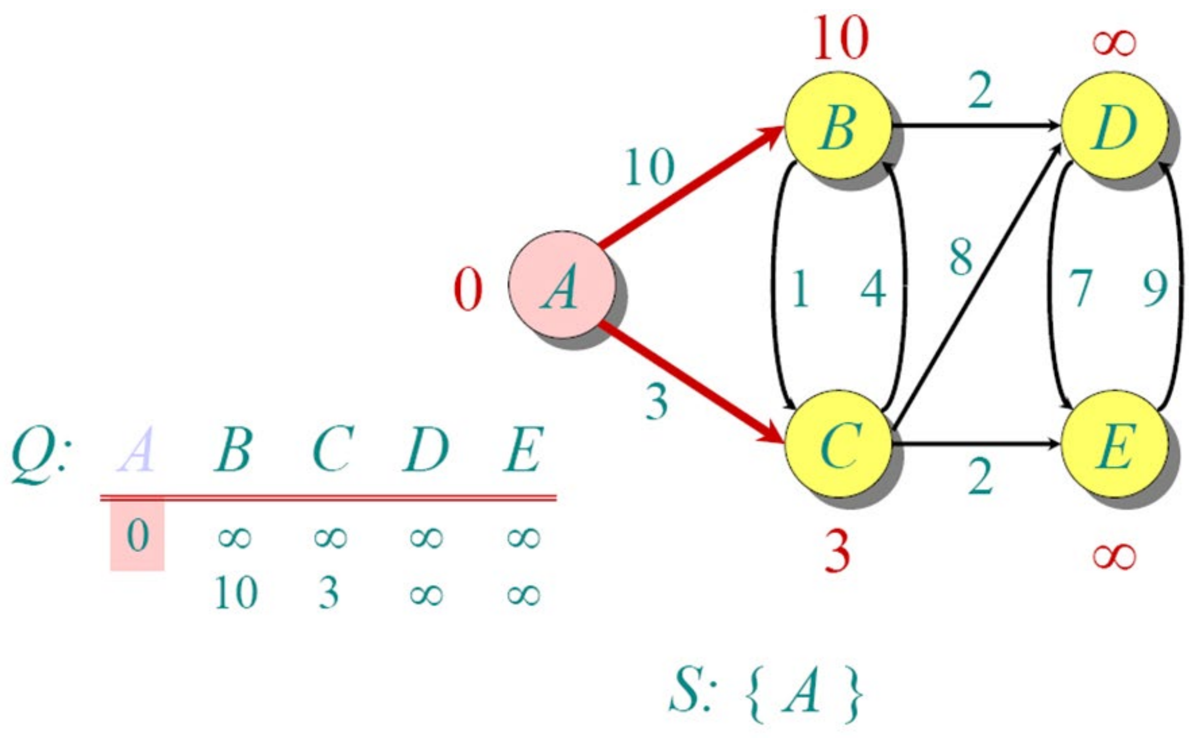
A	B	C	D	E
0	∞	∞	∞	∞

S: {}

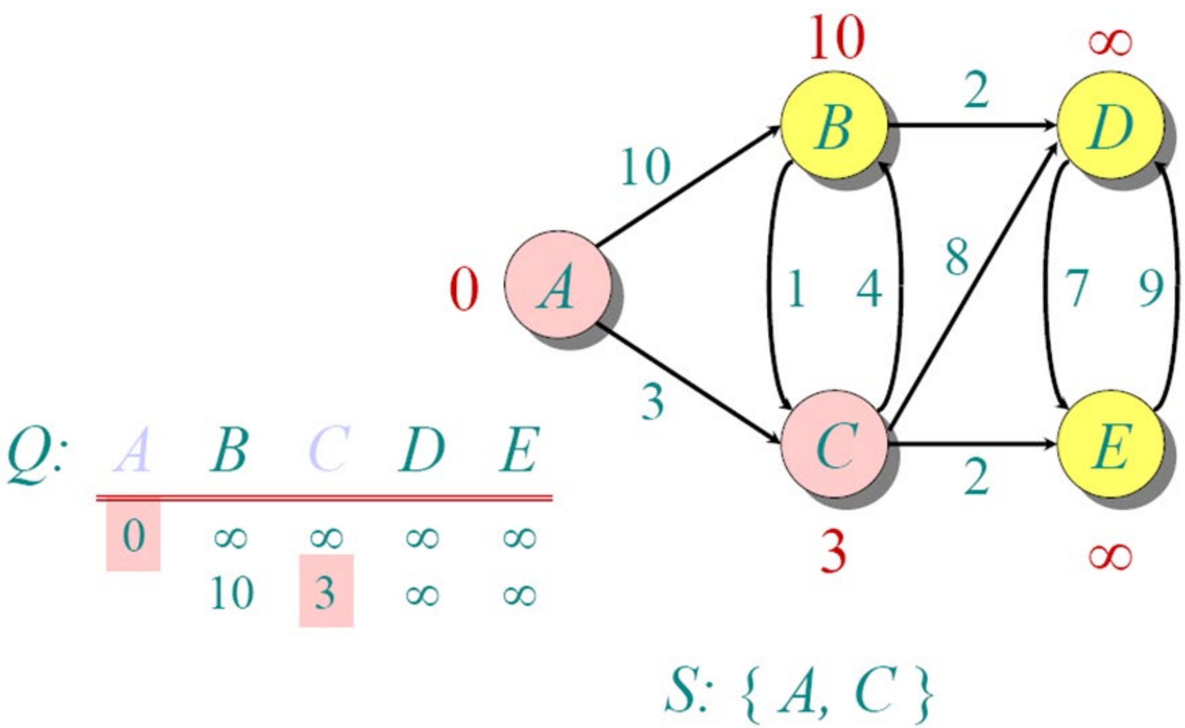
Dijkstra animated example



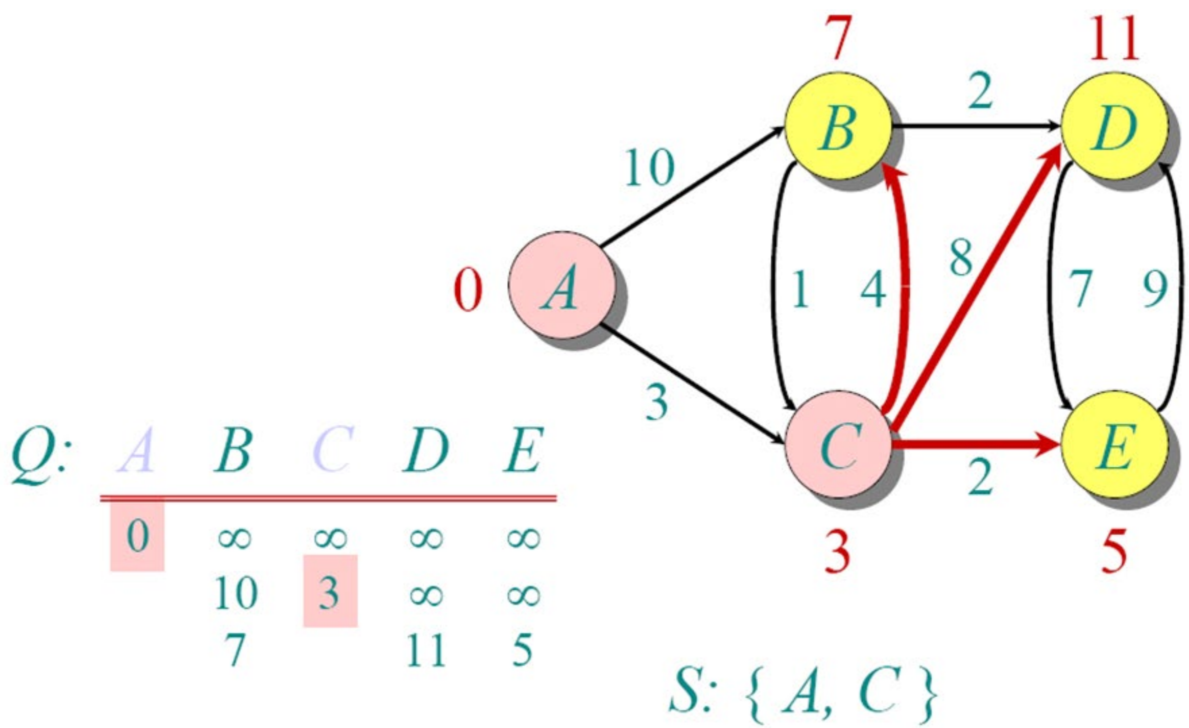
Dijkstra animated example



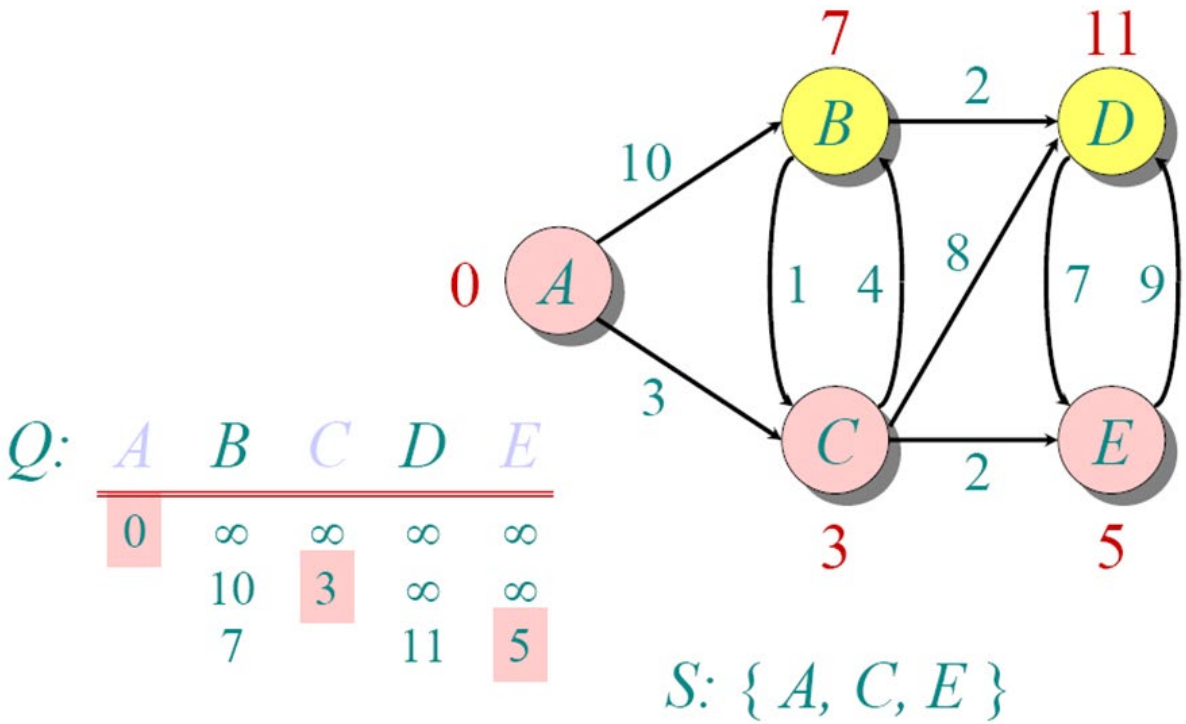
Dijkstra animated example



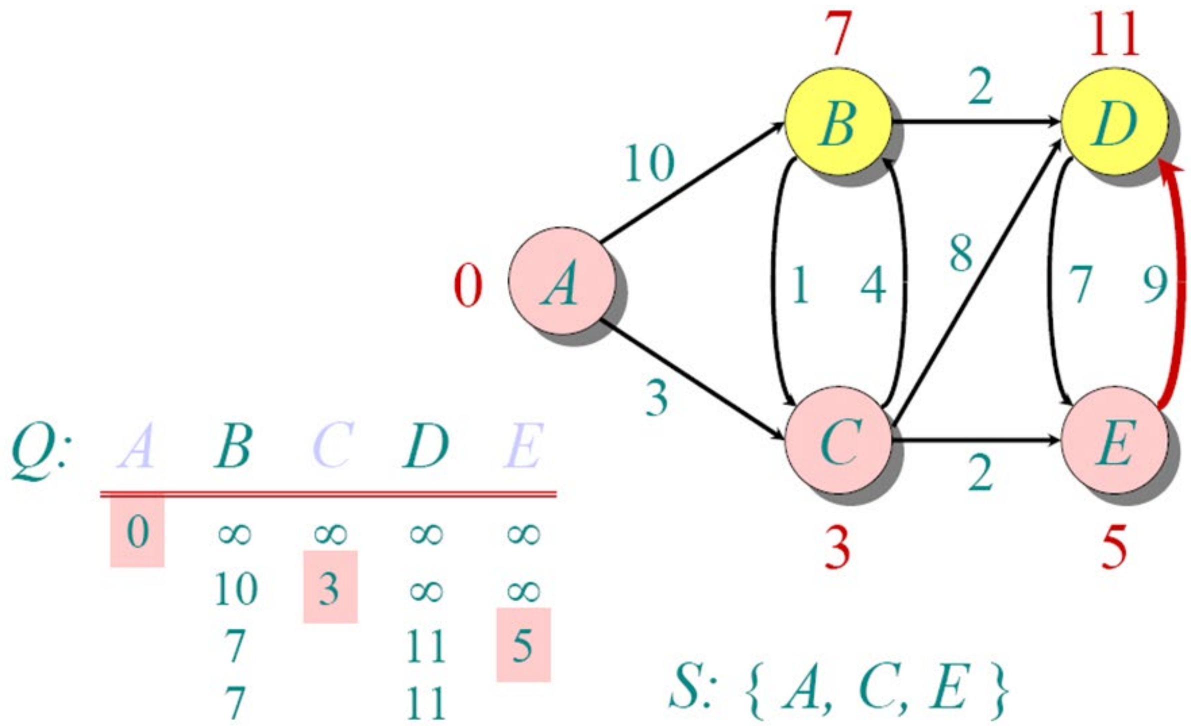
Dijkstra animated example



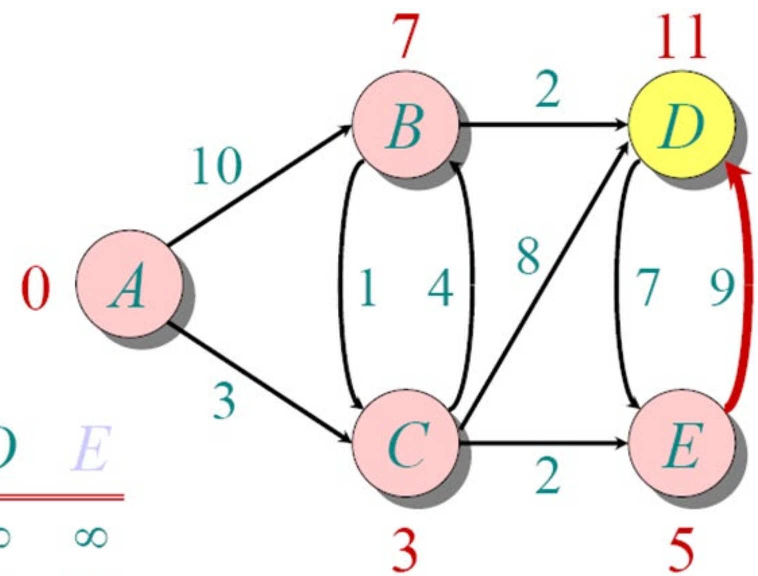
Dijkstra animated example



Dijkstra animated example



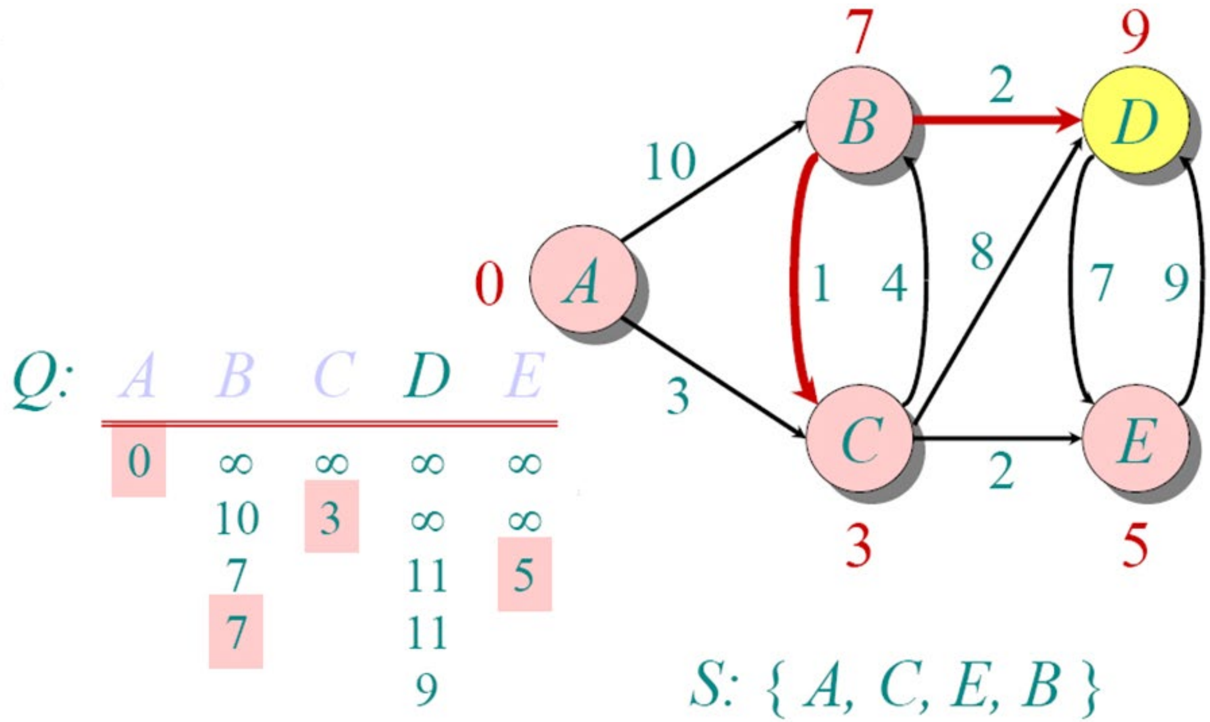
Dijkstra animated example



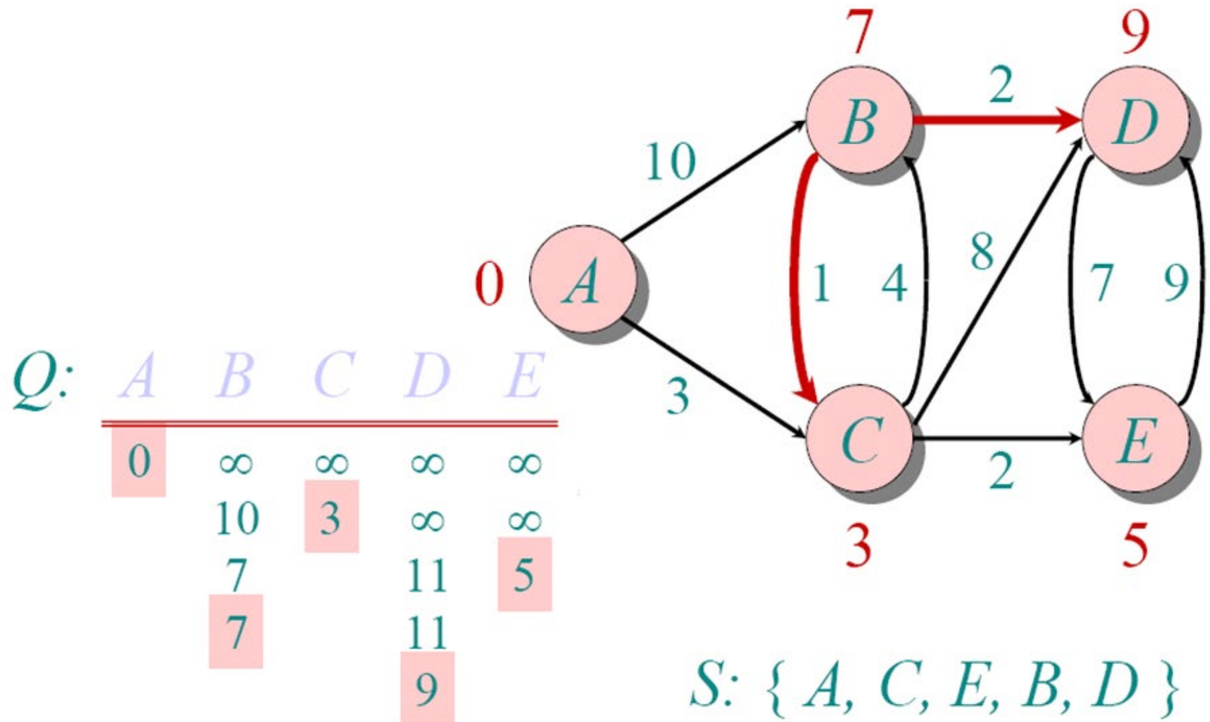
Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	

S: { A, C, E, B }

Dijkstra animated example



Dijkstra animated example



Implementations and running Times

- Simplest implementation: store vertices in an array or linked list.

  The running time is
  $O(|V|^2 + |E|)$

- For sparse graphs: very few edges & many nodes, store graph in an adjacency list using a binary heap or priority queue.

  The running time is
  $O((|E|+|V|) \log |V|)$

Does it works everywhere ?

Bad news:

- No. Doesn’t work when the cost of paths are negative.

Good news:

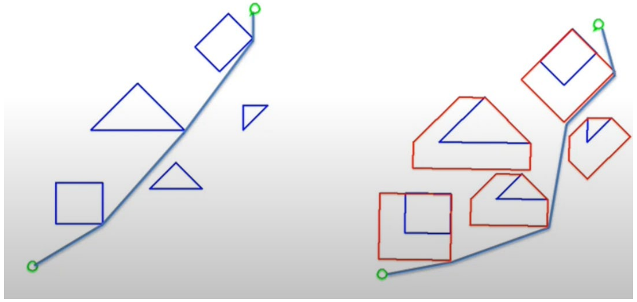
- Build bellman equation and solve it using other methods, e.g., policy/value iteration.

Note: optimal but not unique.

Common questions

Real robot is 3D with some volume, it may collides with obstacle edges.

- Inflate the obstacles.

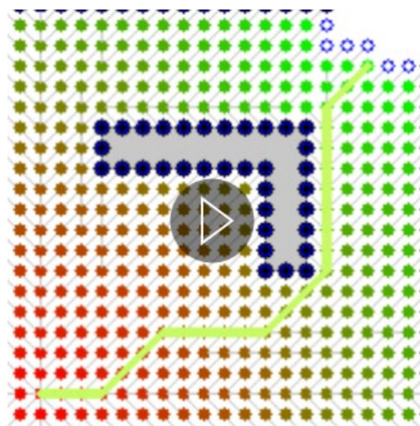


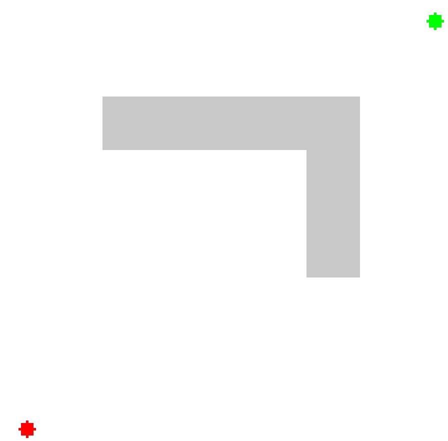
notes: need to find an optimal solution

credit: Edsger W. Dijkstra

- Dutch Computer Scientist
- Received Turing Award for contribution to developing programming languages.

1930-2002





What to learn more?

- Dijkstra's original paper:
  E. W. Dijkstra. (1959) A Note on Two Problems in Connection with Graphs. Numerische Mathematik, 1. 269-271.
- MIT OpenCourseware, 6.046J Introduction to Algorithms.
  < http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-046JFall-2005/CourseHome/> Accessed 4/25/09
- Meyers, L.A. (2007) Contact network epidemiology: Bond percolation applied to infectious disease prediction and control. Bulletin of the American Mathematical Society 44: 63-86.
- Department of Mathematics, University of Melbourne. Dijkstra's Algorithm.
  <http://www.ms.unimelb.edu.au/~moshe/620-261/dijkstra/dijkstra.html > Accessed 4/25/09
- *Introduction to Algorithms* by Cormen, Leiserson and Rivest (MIT Press/McGraw-Hill 1994, ISBN 0-262-03141-8 (MIT Press) and ISBN 0-07-013143-0 (McGraw-Hill).
- http://en.wikipedia.org/wiki/BellmanFord_algorithm
- http://en.wikipedia.org/wiki/Dijkstra_algorithm
- www.Criticalblue.com
- http://www.cs.mcgill.ca/~cs251/OldCourses/1997/topic29/

- *Introduction to Algorithms* by Cormen, Leiserson and Rivest (MIT Press/McGraw-Hill 1994, ISBN 0-262-03141-8 (MIT Press) and ISBN 0-07-013143-0 (McGraw-Hill).