Algorithm Engineering

1. Introduction, Organization & Dates

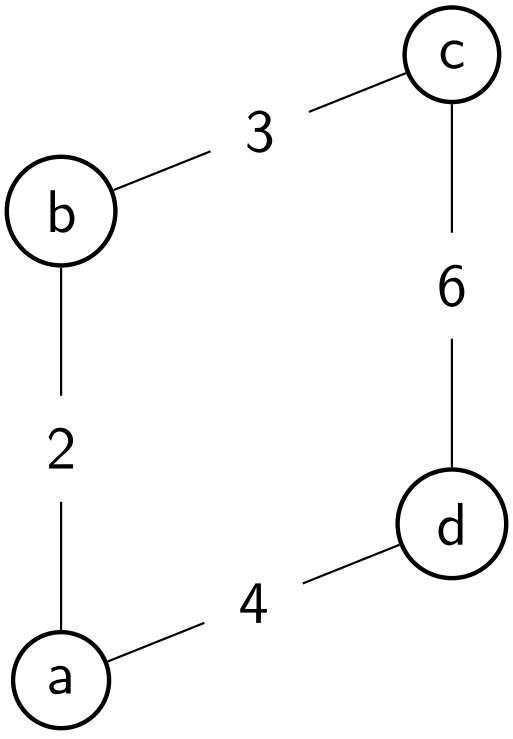
Augusto Blaas Corrêa, Florian Pommerening and Malte Helmert

University of Basel

September 23, 2022

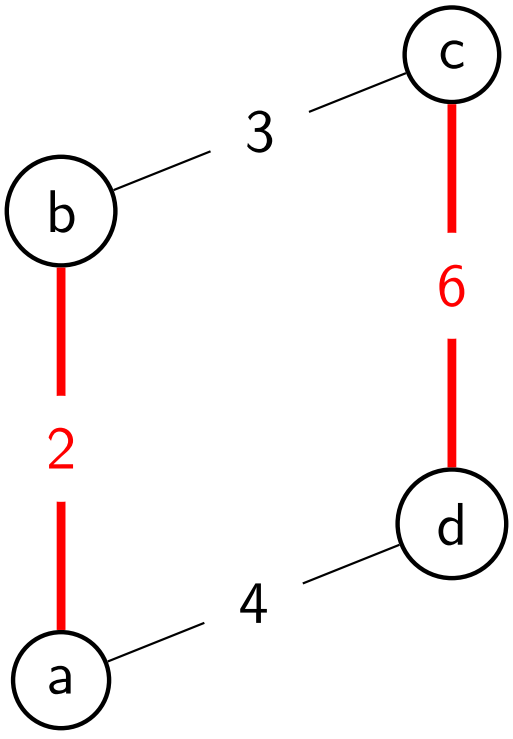
Algorithm Engineering

It's a (Perfect) Match!



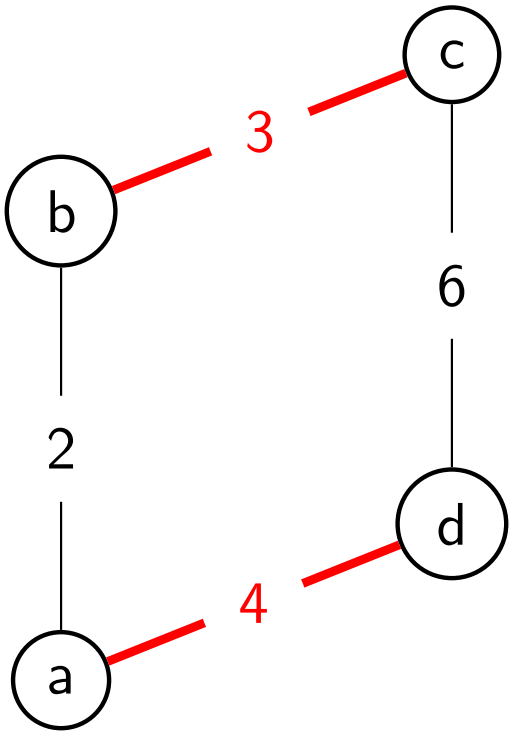
perfect matching: subset of the edges such that every vertex in the graph is connected to exactly one edge of this subset

It's a (Perfect) Match!



perfect matching: subset of the edges such that every vertex in the graph is connected to exactly one edge of this subset

It's a (Perfect) Match!



perfect matching: subset of the edges such that every vertex in the graph is connected to exactly one edge of this subset

perfect matching with minimum cost:
minimize cost of edges in the subset

A Bit of History

1965: Edmonds invents the blossom algorithm, running in $O(V^2E)$

A Bit of History

1965: Edmonds invents the blossom algorithm, running in $O(V^2E)$

1973: Gabow shows an implementation running in $O(V^3)$

A Bit of History

1965: Edmonds invents the blossom algorithm, running in $O(V^2E)$

1973: Gabow shows an implementation running in $O(V^3)$

1986: Galil, Mical, and Gabow improve it to $O(EV \log V)$

A Bit of History

1965: Edmonds invents the blossom algorithm, running in $O(V^2E)$

1973: Gabow shows an implementation running in $O(V^3)$

1986: Galil, Mical, and Gabow improve it to $O(EV \log V)$

1989: Gabow, Galil, and Spencer improve it further to

$$O(EV \log\log\log_{\max(\frac{E}{V},2)} V + V^2 \log V)$$

A Bit of History

1965: Edmonds invents the blossom algorithm, running in $O(V^2E)$

1973: Gabow shows an implementation running in $O(V^3)$

1986: Galil, Mical, and Gabow improve it to $O(EV \log V)$

1989: Gabow, Galil, and Spencer improve it further to

$$O(EV \log\log\log_{\max(\frac{E}{V},2)} V + V^2 \log V)$$

1990: Gabow improves the running time to $O(V(E + \log V))$

...

Blossom Algorithm in Practice

what do people use in practice?

- usually use Blossom V or more modern approaches
- built on Blossom V and its predecessor, Blossom IV

Blossom Algorithm in Practice

what do people use in practice?

- usually use Blossom V or more modern approaches
- built on Blossom V and its predecessor, Blossom IV

running time: Blossom V runs in time $O(V^3E)$

- asymptotically worse than the implementation from 1965
- it is good in practice though...

Theory vs. Practice

asymptotic analysis is a good guidance

- helps us estimate worst/average cases
- tells us how hard a problem is

but practice might be different

- hidden constants
- implementation difficulty
- practical instances are not similar to the worst case

Algorithm Engineering Seminar

problem solving: I give you a problem, how do you solve it?

- think of the nuances of your problem to choose an algorithm
- implement it efficiently while writing good code

goal:

- learn different algorithms for important problems
- understand how to decide which algorithm to use
- have more tools in your toolbox!

Organization

Target Group and Prerequisites

Target Group

- BSc students of Computer Science and similar subjects

Prerequisites

- basic programming skills
- ability to work independently

Capacity Limit

capacity: 20 students in this seminar

priority rules:

- must select one of the available seminars if overbooked
- BSc students that need the credits and registered until September 23
- otherwise first-come first-served

Format

Seminar Format

- theoretical part + programming assignments
- 6 ECTS points
- assessment: graded (1.0–6.0)

Objectives

Objectives

seminar:

- reading and understanding scientific literature
- preparing and presenting scientific talks
- conducting scientific discussions with peers
- writing and discussing scientific reports

programming assignments:

- independent solution of programming problems
- clean and efficient code

Meetings

Sep 23	14:15–16:30	organization, dates & topics
Oct 07	14:15–16:30	seminar presentations
Oct 14	14:15–16:30	lab
Oct 21	14:15–16:30	seminar presentations
Oct 28	14:15–16:30	lab
Nov 04	14:15–16:30	seminar presentations
Nov 11	14:15–16:30	lab
Nov 18	14:15–16:30	seminar presentations
Nov 25	no meeting	Dies Academicus (questions by arrangement)
Dec 02	14:15–16:30	seminar presentations
Dec 09	14:15–16:30	lab
Dec 16	14:15–16:30	final meeting

Labs are not mandatory!

Requirements I

Requirements to pass the course

- write a seminar report
  - 10–12 pages, LaTeX
  - anonymized initial version due: November 6
  - final version due: November 27
- write a peer review
  - 1–2 pages
  - due: November 20

Requirements II

Requirements to pass the course (continued)

- give a seminar presentation
  - 20 minutes presentation (excluding discussion)
  - draft version due: discuss with your supervisor
  - final version due: the evening before the talk

Requirements III

Requirements to pass the course (continued)

- submit an implementation for all programming assignments
  - due: the evening before the next topic presentation
  - i.e., around two weeks per assignment
- participate in all meetings
- actively participate in discussions

Grading

Grading

- written report (final version) (25%)
- peer review (15%)
- seminar presentation (25%)
- programming assignments (35%)

Each component is graded individually on a scale of 1.0–6.0.
The final grade is the weighted average of all component grades.

Peer Review

- You will receive someone else's seminar report and give written feedback.
- Feedback is anonymous.
- Goal: learn through new perspective

Programming Assignments

one assignment per topic

- use Kattis online judge
- submit your code, Kattis runs several tests
- evaluate correctness and performance

each assignment also has a competition

- does not influence grade
- metric depends on the problem (usually performance)

University of Basel Kattis

We have our own Kattis website: `https://unibas.kattis.com/`

- find assignments, submit your code, check your score

Competition

we also have a programming competition:

- every week, we rank the assignments based on their efficiency
- more efficient $\Longrightarrow$ more points in the competition
- top 3 places per week get chocolate
- top 5 places over all weeks get a final prize at the last meeting

note: performance in the competition does not impact the grade

Organizers and Supervisors

Augusto B. Corrêa

Florian Pommerening

Malte Helmert

Liat Cohen

Salomé Eriksson

Thomas Keller

Clemens Büchner

Remo Christen

Simon Dold

Patrick Ferber

Organizers

Organizers

Augusto B. Corrêa (main contact)

- email: augusto.blaascorrea@unibas.ch
- office: Spiegelgasse 5, room 04.001

Florian Pommerening

- email: florian.pommerening@unibas.ch
- office: Spiegelgasse 1, room 04.005

Malte Helmert

- email: malte.helmert@unibas.ch
- office: Spiegelgasse 1, room 06.004

Other Supervisors

- Spiegelgasse 1, room 04.002
  - Salomé Eriksson (salome.eriksson@unibas.ch)
- Spiegelgasse 1, room 04.005
  - Thomas Keller (tho.keller@unibas.ch)
- Spiegelgasse 5, room 04.001
  - Clemens Büchner (clemens.buechner@unibas.ch)
  - Remo Christen (remo.christen@unibas.ch)
  - Liat Cohen (liat.cohen@unibas.ch)
  - Simon Dold (simon.dold@unibas.ch)
  - Patrick Ferber (patrick.ferber@unibas.ch)

Course Links I

Seminar Homepage

`https://dmi.unibas.ch/en/studies/computer-science/courses-in-fall-semester-2022/seminar-algorithm-engineering/`

- seminar description
- slides
- examples for good talks, reports, and reviews

Course Links II

Discord Server

- place to interact with us and with each other

Kattis

`https://unibas.kattis.com/`

- submission of programming assignments

ADAM Workspace

`https://adam.unibas.ch/goto_adam_crs_1374662.html`

- link to Discord and Kattis
- recorded lectures by course organizers

Plagiarism

Plagiarism

- plagiarism: representing work or ideas of other people as your own
- consequence: failing the seminar
- if in doubt: ask us in advance!

Repeat offenders can be excluded from the study program.

Language

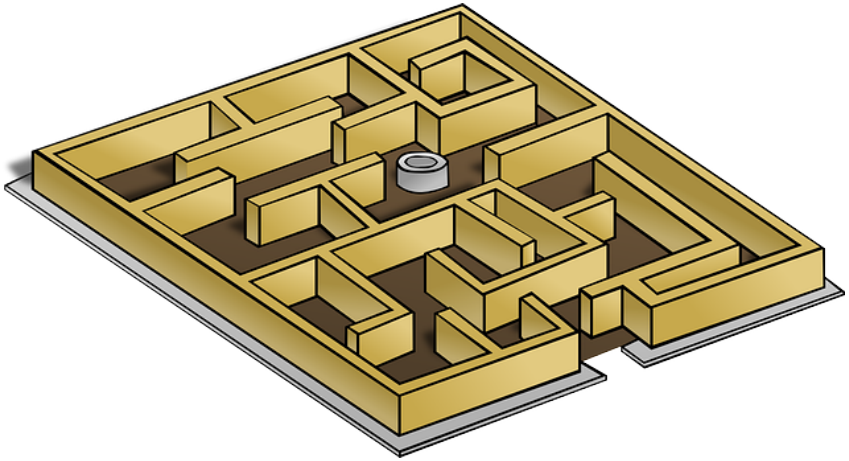
- The seminar is in English (reports, presentations, etc.).
- If you prefer a supervisor who speaks German, let us know.

Questions about the Organization

Questions?

Topics

Path-finding



find a path in a graph

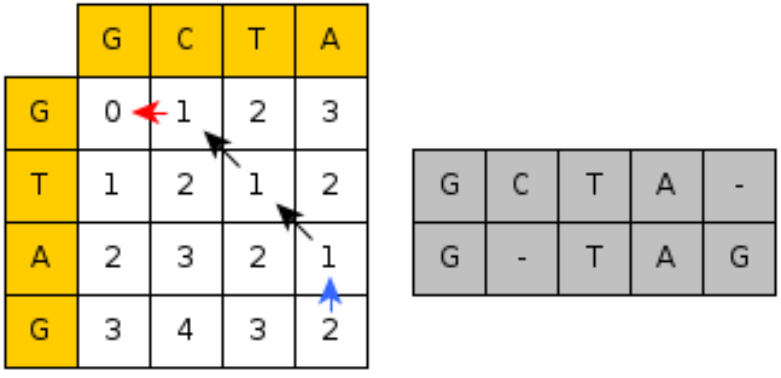
- depends on the type of graph and specific problem
- between two specific nodes vs. between any two nodes
- informed search vs. uninformed search
- arbitrary graph vs. structured graph (e.g., grid)

Network Flow

find maximum flow passing through a graph

- several algorithms: Edmonds-Karp, Push-Relabel, etc.
- used also to compute minimum graph cut and matching

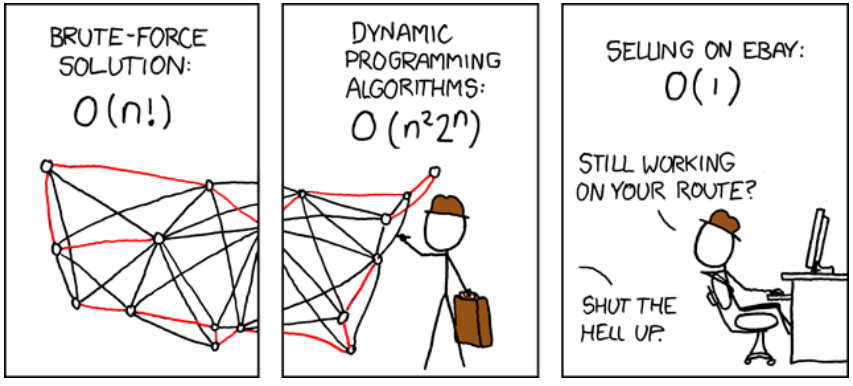
Dynamic Programming



technique used in several different problems

- subset sum and knapsack problems
- sequence alignment

Traveling Salesperson Problem (TSP)

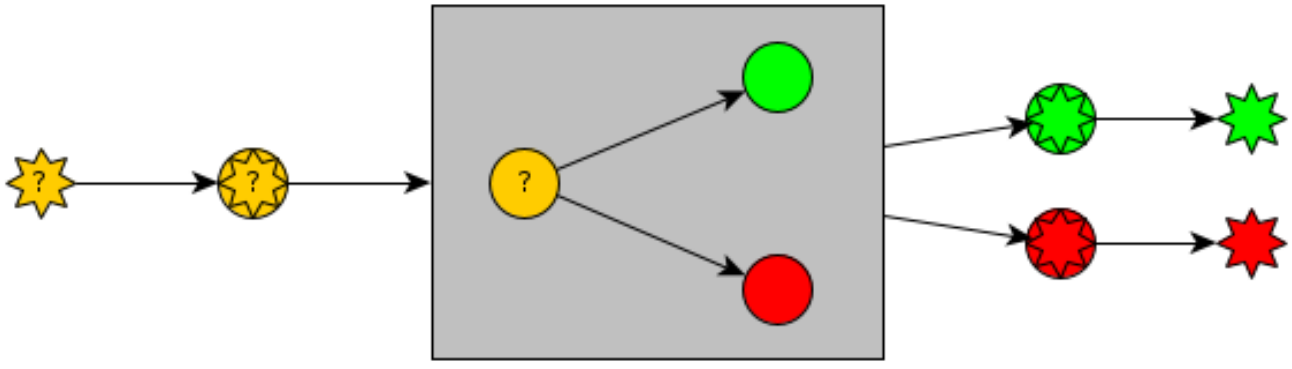


https://xkcd.com/399

find shortest tour visiting each node exactly once

- illustrate several techniques:
  - randomized algorithms
  - local search
  - approximation algorithms

Solving Problems Via Reductions



reduce your problem to another one

- easier to use an existing solver than to create your own
- reduce to SAT, CSP, MIP, etc.

Next Steps

Topic Assignment

until tonight (September 23):

- send us the topics ordered from most preferred to least preferred

$\rightsquigarrow$ We will send out the topics, supervisor assignments, and presentation dates tomorrow (Saturday).

If you don't send preferences, we assume that you don't intend to participate in the seminar.

Next Steps

- Create a Kattis account and solve the first problem.
  - Sign up at `https://unibas.kattis.com/` with your university email address.
  - Once logged in, click on the seminar and then "hs22".
  - Register as a student (find the key on ADAM).
  - Solve assignment "Aaah!" until October 7.
- Contact your supervisor early and schedule meetings.
- Start reading the material as early as possible.

Important Dates

Sep 23	due date: topic suggestions
Sep 24	topic assignment
Oct 07	meeting: seminar presentations
Oct 14	meeting: lab
Oct 21	meeting: seminar presentations
Oct 28	meeting: lab
Nov 04	meeting: seminar presentations
Nov 06	due date: initial report version
Nov 11	meeting: lab
Nov 18	meeting: seminar presentations
Nov 20	due date: peer review
Nov 25	no meeting: Dies Academicus (questions by arrangement)
Nov 27	due date: final report version
Dec 2	meeting: seminar presentations
Dec 9	meeting: lab
Dec 16	meeting: final meeting