CS 577 - Introduction to Algorithms

Manolis Vlatakis

Department of Computer Sciences
University of Wisconsin – Madison

Fall 2024

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

CS 577 - Introduction to Algorithms: Fall 2024

About Me

University of California, Berkeley

2022-2024

Simons Institute For Theory Of Computing

Post-doctoral studies

Columbia University

2016-2022

Department Of Computer Sciences

Graduate & doctorate studies

National Technical University of Athens

2010-2016

School Of Electrical And Computer Engineering

Undergraduate studies

Manolis Vlatakis

Emmanouil V. Vlatakis Gkaragkounis

About You

My current year in school is:

a. Freshman

b. Sophomore

c. Junior

d. Senior

e. Graduate Student

f. Other

About You

My primary reason for taking CS 577:

a. I am very interested in the subject.

b. I am curious to learn more about the subject.

c. It fulfils a requirement for my program, major or certificate.

d. It fits my schedule.

e. I’ve heard good things about the course.

About You

My favorite Harry Potter book/movie is:

a. Harry Potter and the Philosopher's Stone
b. Harry Potter and the Chamber of Secrets
c. Harry Potter and the Prisoner of Azkaban
d. Harry Potter and the Goblet of Fire
e. Harry Potter and the Order of the Phoenix
f. Harry Potter and the Half-Blood Prince
g. Harry Potter and the Deathly Hallows: Part 1
h. Harry Potter and the Deathly Hallows: Part 2
i. Never read/seen them

What is the Focus of This Course?

Tackling "Challenging" Problems

□ What do we do when a problem seems “difficult”?

What is the Focus of This Course?

Tackling "Challenging" Problems

□ What do we do when a problem seems “difficult”?

We use all CS 577 tricks

AND

we solve the problem

What is the Focus of This Course?

Tackling "Challenging" Problems

- □ What do we do when a problem seems “difficult”?
  - ■ “Easy”: After significant effort, we find an efficient algorithm (polynomial-time).
  - ■ “Difficult”: After significant effort, we cannot find an efficient algorithm (polynomial-time).
- □ We go to the boss and say:

What is the Focus of This Course?

Tackling "Challenging" Problems

- What do we do when a problem seems “difficult”?
  - “Easy”: After significant effort, we find an efficient algorithm (polynomial-time).
  - “Difficult”: After significant effort, we cannot find an efficient algorithm (polynomial-time).
- We go to the boss and say:
  - I can’t find an efficient algorithm. You’re fired!

What is the Focus of This Course?

Tackling "Challenging" Problems

- □ What do we do when a problem seems “difficult”?
  - ■ “Easy”: After significant effort, we find an efficient algorithm (polynomial-time).
  - ■ “Difficult”: After significant effort, we cannot find an efficient algorithm (polynomial-time).
- □ We go to the boss and say:
  - ■ I can’t find an efficient algorithm. You’re fired!
  - ■ There isn’t an efficient algorithm. Good, but difficult!

What is the Focus of This Course?

Tackling "Challenging" Problems

- □ What do we do when a problem seems “difficult”?
  - ■ “Easy”: After significant effort, we find an efficient algorithm (polynomial-time).
  - ■ “Difficult”: After significant effort, we cannot find an efficient algorithm (polynomial-time).
- □ We go to the boss and say:
  - ■ I can’t find an efficient algorithm. You’re fired!
  - ■ There isn’t an efficient algorithm. Good, but difficult!
  - ■ No one can find an efficient algorithm (and everyone believes it doesn’t exist).
- □ Theory of NP-completeness.
  - ■ NP-complete: A class of extremely important problems where either all of them are solved in polynomial time or none of them are.

Analysis of Algorithms

Problem

- Mathematical model of the problem area.
- Rules of the game.

- **Sorting:** Given a list of n numbers, rearrange them in increasing (or decreasing) order.
- **Searching:** Given a sorted array of n elements, find the position of a target value.
- **Shortest Path:** Given a graph with weighted edges, find the shortest path between two vertices.
- **Dynamic Programming:** Compute the population of rabbits in k-epochs (Fibonacci sequence)
- **Graph Traversal:** Explore nodes and edges of a graph in a systematic way.

Analysis of Algorithms

Problem

- Mathematical model of the problem area.
- Rules of the game.

Algorithm

- Step-by-step procedure for solving an instance of a given problem.

- **Sorting:** Given a list of n numbers, rearrange them in increasing (or decreasing) order.
- **Searching:** Given a sorted array of n elements, find the position of a target value.
- **Shortest Path:** Given a graph with weighted edges, find the shortest path between two vertices.
- **Dynamic Programming:** Compute the population of rabbits in k-epochs (Fibonacci sequence)
- **Graph Traversal:** Explore nodes and edges of a graph in a systematic way.

Analysis of Algorithms

Standard Cooking Anecdote ☺

Problem

- Mathematical model of the problem area.
- Rules of the game.

Algorithm

- Step-by-step procedure for solving an instance of a given problem.

- Ex: I have kitchen with a stocked pantry and I want a cookie.

Analysis of Algorithms

Standard Cooking Anecdote ☺

Problem

- Mathematical model of the problem area.
- Rules of the game.

Algorithm

- Step-by-step procedure for solving an instance of a given problem.

- Ex: I have kitchen with a stocked pantry and I want a cookie.

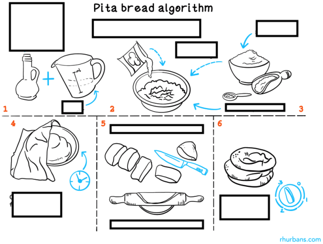
- Ex: Given a kitchen with a stove, etc... and a pantry with chocolate chips, etc...

Algorithms Vs Code

Algorithm

- Step-by-step procedure for solving an instance of a given problem.

- Ex: Use basic ingredients and tools...

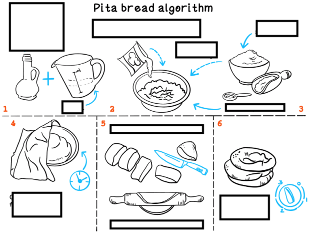


Algorithms Vs Code

Algorithm

- Step-by-step procedure for solving an instance of a given problem.

- Ex: Use basic ingredients and tools...

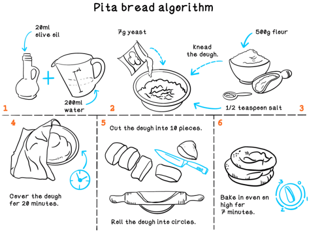


rhurbans.com

Code

- Formal step-by-step procedure for solving an instance of a given problem.

- Ex: Given a kitchen with detailed instructions...


Pita bread algorithm
20ml olive oil
200ml water
1
7g yeast
2
Knead the dough.
500g flour
1/2 teaspoon salt
3
4
Cover the dough for 20 minutes.
5
Cut the dough into 10 pieces.
Roll the dough into circles.
6
Bake in oven on high for 7 minutes.

rhurbans.com

Stable Matching Problem (SMP) (1962)[1][2][3]

Problem Definition

Given a set of n hospitals, H, and a set of n doctors, D. Each hospital and doctor has a preference ranking of the opposite set. Compute a stable matching between H and D. A matching is stable if it is (i) perfect, and (ii) there are no pairs (h, d) and (h', d') in the matching where h prefers d' and d' prefers h.

[1] Algorithm Design, Ch 1.

[2] Algorithms, Ch 4.5

[3] `http://mathsite.math.berkeley.edu/smp/smp.html` (Uses Flash)

Stable Matching Problem (SMP) (1962)[1][2][3]

Problem Definition

Given a set of n hospitals, H, and a set of n doctors, D. Each hospital and doctor has a preference ranking of the opposite set. Compute a stable matching between H and D. A matching is stable if it is (i) perfect, and (ii) there are no pairs (h, d) and (h', d') in the matching where h prefers d' and d' prefers h.

- Also known as the Stable Marriage Problem.
  (Historically, in the '60s framed as matching men & women.)
- There are more complex variations of the model.
- Used in the real world (e.g., matching students to schools).
- Nobel Prize in Economics (2012) Shapley & Roth.

[1] Algorithm Design, Ch 1.

[2] Algorithms, Ch 4.5

[3] `http://mathsite.math.berkeley.edu/smp/smp.html` (Uses Flash)

Gale-Shapely Algorithm [4] for SMP (1962)

```
Initially all $m \in M$ and $w \in W$ are free
while there is a man $m$ who is free and hasn't proposed to every woman do
    Choose such a man $m$
    Let $w$ be the highest-ranked woman in $m$'s preference list to whom $m$ has not yet proposed
    if $w$ is free then
        $(m, w)$ become engaged
    else $w$ is currently engaged to $m'$
        if $w$ prefers $m'$ to $m$ then
            $m$ remains free
        else $w$ prefers $m$ to $m'$
            $(m, w)$ become engaged
            $m'$ becomes free
        end
    end
end
return the set $S$ of engaged pairs
```

[4] Algorithm Design, p.6

Gale-Shapely Algorithm [4] for SMP (1962)

```
Initially all $m \in M$ and $w \in W$ are free
while there is a man $m$ who is free and hasn't proposed to every woman do
    Choose such a man $m$
    Let $w$ be the highest-ranked woman in $m$'s preference list to whom $m$ has not yet proposed
    if $w$ is free then
        $(m, w)$ become engaged
    else $w$ is currently engaged to $m'$
        if $w$ prefers $m'$ to $m$ then
            $m$ remains free
        else $w$ prefers $m$ to $m'$
            $(m, w)$ become engaged
            $m'$ becomes free
        end
    end
end
return the set $S$ of engaged pairs
```

Is it good?

[4] Algorithm Design, p.6

Gale-Shapely Algorithm [4] for SMP (1962)

```
Initially all $m \in M$ and $w \in W$ are free
while there is a man $m$ who is free and hasn't proposed to every woman do
    Choose such a man $m$
    Let $w$ be the highest-ranked woman in $m$'s preference list to whom $m$ has not yet proposed
    if $w$ is free then
        $(m, w)$ become engaged
    else $w$ is currently engaged to $m'$
        if $w$ prefers $m'$ to $m$ then
            $m$ remains free
        else $w$ prefers $m$ to $m'$
            $(m, w)$ become engaged
            $m'$ becomes free
        end
    end
end
return the set $S$ of engaged pairs
```

Is it good?

- Complete?

[4] Algorithm Design, p.6

Gale-Shapely Algorithm [4] for SMP (1962)

```
Initially all $m \in M$ and $w \in W$ are free
while there is a man $m$ who is free and hasn't proposed to every woman do
    Choose such a man $m$
    Let $w$ be the highest-ranked woman in $m$'s preference list to whom $m$ has not yet proposed
    if $w$ is free then
        $(m, w)$ become engaged
    else $w$ is currently engaged to $m'$
        if $w$ prefers $m'$ to $m$ then
            $m$ remains free
        else $w$ prefers $m$ to $m'$
            $(m, w)$ become engaged
            $m'$ becomes free
        end
    end
end
return the set $S$ of engaged pairs
```

Is it good?

- Complete?
- Correct?

[4] Algorithm Design, p.6

Gale-Shapely Algorithm [4] for SMP (1962)

```
Initially all $m \in M$ and $w \in W$ are free
while there is a man $m$ who is free and hasn't proposed to every woman do
    Choose such a man $m$
    Let $w$ be the highest-ranked woman in $m$'s preference list to whom $m$ has not yet proposed
    if $w$ is free then
        $(m, w)$ become engaged
    else $w$ is currently engaged to $m'$
        if $w$ prefers $m'$ to $m$ then
            $m$ remains free
        else $w$ prefers $m$ to $m'$
            $(m, w)$ become engaged
            $m'$ becomes free
        end
    end
end
return the set $S$ of engaged pairs
```

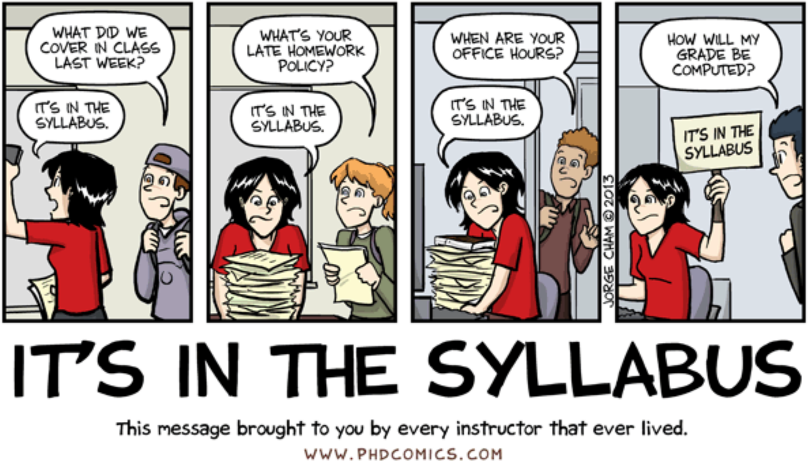
Is it good?

- Complete?
- Correct?
- Efficient? With respect to what (time, space, ...)?

[4] Algorithm Design, p.6

Syllabus (Course Logistics)

https://canvas.wisc.edu/courses/429678


WHAT DID WE COVER IN CLASS LAST WEEK?
IT'S IN THE SYLLABUS.
WHAT'S YOUR LATE HOMEWORK POLICY?
IT'S IN THE SYLLABUS.
WHEN ARE YOUR OFFICE HOURS?
IT'S IN THE SYLLABUS.
JORGE CHAM © 2013
HOW WILL MY GRADE BE COMPUTED?
IT'S IN THE SYLLABUS
IT'S IN THE SYLLABUS
This message brought to you by every instructor that ever lived.
WWW.PHDCOMICS.COM

Course Aim

https://canvas.wisc.edu/courses/429678

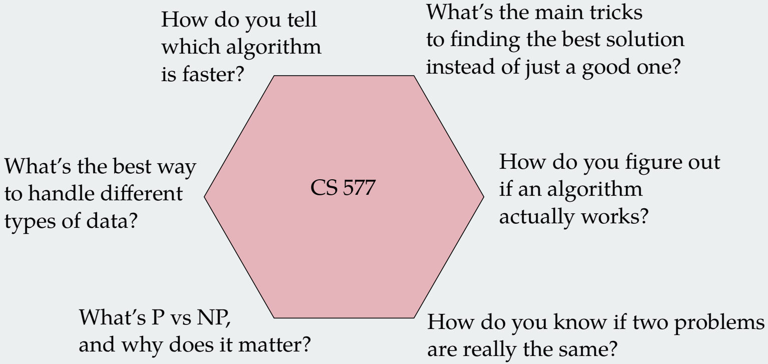
Overall

- Basic paradigms for the design and analysis of efficient algorithms:
  - greedy,
  - divide-and-conquer,
  - dynamic programming,
  - reductions, and
  - the use of randomness.
- Computational intractability including typical NP-complete problems and ways to deal with them.

Course Aim

HTTPS://CANVAS.WISC.EDU/COURSES/429678

Specific Learning Outcomes



CS 577

How do you tell which algorithm is faster?

What's the main tricks to finding the best solution instead of just a good one?

What's the best way to handle different types of data?

How do you figure out if an algorithm actually works?

What's P vs NP, and why does it matter?

How do you know if two problems are really the same?

Getting Started

Getting Started Checklist

https://canvas.wisc.edu/courses/429678

Checklist

1. Review the Syllabus
2. Activate Piazza account
3. Register for Gradescope
4. Exam Conflicts

1. Review the Syllabus

Grading

- Bonus credits (up to 10%). In the assignments and later in the semester, I will offer a fun challenge for extra bonus points.

Bonus Credits

1. Review the Syllabus

Grading

- Midterm Exam (25%)
  - Monday, October 17, 2024 @ In-class
  - Pretty Easy ☺

1. Review the Syllabus

Grading

- Midterm Exam (25%)
- Assignments (42%)
  - 4-7 assignments in total (due Tuesdays 23:59)
  - Graded also on participation, not only correctness!
  - Participation credit requires a reasonable attempt to answer a question.

1. Review the Syllabus

Grading

- Midterm Exam (25%)
- Assignments (42%)
- Final Exam (33%)
  - Monday, December 16, 2024 @ 10:05AM - 12:05PM

1. Review the Syllabus

Course Expectations

Doing less than 70% of the assigned discussions, classes, and assignments risks altering the knowledge and skills of the course, lowering the academic standards, and fundamentally altering the nature of the course.

- We expect every student to attend lectures, discussions, submit all homework, and complete all quizzes.
- The flexibility is provided because life happens, NOT because we expect students to only do 70% of the work or skip quizzes.

1. Review the Syllabus

Academic Integrity

- Academic dishonesty or misconduct is taken very seriously by the university (see UW–Madison Academic Integrity policy).

1. Review the Syllabus

Academic Integrity

- Academic dishonesty or misconduct is taken very seriously by the university (see UW–Madison Academic Integrity policy).
- It is academic misconduct to submit someone else's work as your own.
- It is academic misconduct to help another student commit academic misconduct.

1. Review the Syllabus

Academic Integrity

- Academic dishonesty or misconduct is taken very seriously by the university (see UW–Madison Academic Integrity policy).
- It is academic misconduct to submit someone else's work as your own.
- It is academic misconduct to help another student commit academic misconduct.

Peer Help on Assignments

- You may not email, post on Piazza, or otherwise make solutions (or parts of them) available to others.
- Process:
  - If you receive or give help on an assignment, be sure to cite the person who helped.

2. Activate Piazza Account

piazza

https://piazza.com/wisc/fall2024/cs577

Online question resource

- One discussion area for all sections.
- Interaction of students, TAs and instructor.
- First stop for getting questions answered.

2. Activate Piazza Account

piazza

https://piazza.com/wisc/fall2024/cs577

Online question resource

- One discussion area for all sections.
- Interaction of students, TAs and instructor.
- First stop for getting questions answered.

Rules

- Be courteous.
- Don't post answers to homework!
- Search first, post second.

3. Register for Gradescope

How to Register

1. Go to: `https://www.gradescope.com/`
2. The entry code is XGEY74.
3. Use your official wisc.edu exmail address (no aliases)!

3. Register for Gradescope

How to Register

1. Go to: `https://www.gradescope.com/`
2. The entry code is XGEY74.
3. Use your official wisc.edu exmail address (no aliases)!

Submission and Grading Policy

1. For each assignment, you will upload a PDF of the assignment (and code if there is a coding portion).
2. Late Policy:

$$G = \text{Points Earned} \times \text{Max}\left(\frac{\text{ReLU}\,(120 - CH)}{120 - CH}, \frac{(100 - H)}{100}, 0.50\right)$$

4. Exam Conflicts and Accommodations

Conflicts and Accommodations

- Monday, December 16, 2024 @ 10:05AM - 12:05PM
- No later than two weeks before the exam date: Provide your conflicts or accommodations into the following Google form:

  `https://forms.gle/NicgyGNzPFJr6Xcy9`

Textbooks (optional)

- **Kleinberg, and Tardos.** ***Algorithm Design.*** **Addison Wesley, 2006.** Main textbook for 577.

Textbooks (optional)

- **Kleinberg, and Tardos.** ***Algorithm Design.*** **Addison Wesley, 2006.** Main textbook for 577.
- **Jeff Erickson.** ***Algorithms.*** Free online algorithms textbook. jeffe.cs.illinois.edu/teaching/algorithms/

Textbooks (optional)

- **Kleinberg, and Tardos.** ***Algorithm Design.*** **Addison Wesley, 2006.** Main textbook for 577.
- **Jeff Erickson.** ***Algorithms.*** Free online algorithms textbook. jeffe.cs.illinois.edu/teaching/algorithms/
- **Cormen, Leiserson, Rivest, and Stein.** ***Introduction to Algorithms, 3rd Edition.*** **MIT Press, 2009.** The classic ☺

Textbooks (optional)

- **Kleinberg, and Tardos.** ***Algorithm Design.*** **Addison Wesley, 2006.** Main textbook for 577.
- **Jeff Erickson.** ***Algorithms.*** Free online algorithms textbook. jeffe.cs.illinois.edu/teaching/algorithms/
- **Cormen, Leiserson, Rivest, and Stein.** ***Introduction to Algorithms, 3rd Edition.*** **MIT Press, 2009.** The classic ☺
- **Sedgewick, and Wayne.** ***Algorithms, 4th Edition*** **Pearson, 2011.** Another classic textbook with working Java code.

Textbooks (optional)

- **Kleinberg, and Tardos.** ***Algorithm Design.*** **Addison Wesley, 2006.** Main textbook for 577.
- **Jeff Erickson.** ***Algorithms.*** Free online algorithms textbook. jeffe.cs.illinois.edu/teaching/algorithms/
- **Cormen, Leiserson, Rivest, and Stein.** ***Introduction to Algorithms, 3rd Edition.*** **MIT Press, 2009.** The classic ☺
- **Sedgewick, and Wayne.** ***Algorithms, 4th Edition*** **Pearson, 2011.** Another classic textbook with working Java code.
- **Dasgupta, Papadimitriou, and Vazirani.** *Algorithms.* McGraw-Hill Education, 2006. A comprehensive and accessible introduction to algorithms, with a focus on conceptual understanding.

Textbooks (optional)

- **Kleinberg, and Tardos.** ***Algorithm Design.*** **Addison Wesley, 2006.** Main textbook for 577.
- **Jeff Erickson.** ***Algorithms.*** Free online algorithms textbook. jeffe.cs.illinois.edu/teaching/algorithms/
- **Cormen, Leiserson, Rivest, and Stein.** ***Introduction to Algorithms, 3rd Edition.*** **MIT Press, 2009.** The classic ☺
- **Sedgewick, and Wayne.** ***Algorithms, 4th Edition*** **Pearson, 2011.** Another classic textbook with working Java code.
- **Dasgupta, Papadimitriou, and Vazirani.** *Algorithms.* McGraw-Hill Education, 2006. A comprehensive and accessible introduction to algorithms, with a focus on conceptual understanding.
- **Levitin.** *Introduction to the Design and Analysis of Algorithms, 3rd Edition.* Addison Wesley, 2011. A well-structured and pedagogically driven textbook that offers a wide range of algorithmic strategies.

Getting Help

Getting Help

https://canvas.wisc.edu/courses/429678

Help!

- Piazza Online Discussion
- TA Office Hours
- Instructor Office Hours

Appendix

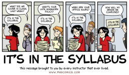
References

Image Sources I

https://piazza.com/

https://brand.wisc.edu/web/logos/

http://bigpicture.typepad.com/comments/images/2008/07/14/dont_panic.png


IT'S IN THE SYLLABUS
This message brought to you by every instructor that ever lived.
WWW.PHDCOMICS.COM

http://phdcomics.com/comics.php?f=1583

https://www.linkedin.com/company/gradescope/