CSE 421: Introduction to Algorithms

Yin-Tat Lee

Blue strip means not on HWs/exams

Quiz

Edit Distance in Near-Linear Time: it's a Constant Factor*

Alexandr Andoni
Columbia University
andoni@cs.columbia.edu

Negev Shekel Nosatzki
Columbia University
ns3049@columbia.edu

May 18, 2020

Abstract

We present an algorithm for approximating the edit distance between two strings of length n in time $n^{1+\epsilon}$, for any $\epsilon > 0$, up to a constant factor. Our result completes the research direction set forth in the recent breakthrough paper [CDG+18], which showed the first constant-factor approximation algorithm with a (strongly) sub-quadratic running time. Several recent results have shown near-linear complexity under different restrictions on the inputs (eg, when the edit distance is close to maximal, or when one of the inputs is pseudo-random). In contrast, our algorithm obtains a constant-factor approximation in near-linear running time for any input strings.

See arXiv: 2005.07678

Input: two n-size strings s_1 and s_2.

- $s_1 = AGGCTACC$
- $s_2 = CAGGCTAC$

Output: minimum number of insertions/deletions to transform s_1 into s_2.

Algorithm: ????

If you forget about Big-Oh notation, don't worry. Next lecture.

Naively, $O(1)^n$ by enumerating all changes.

After this quarter, you should be able to

- solve it in $O(n^2)$ time
- explain why your algorithm is correct.

Question: Is $O(n^{1.999})$ possible?

This Course

Solve computational problems using less steps.

Goal:

- Learn some techniques to design algorithms
- Understand some problems are difficult
- Prove some correctness

Grading Scheme

- Homework 50%
  Weekly homework due on Wed before the class
- Midterm 20%
- Final 30%

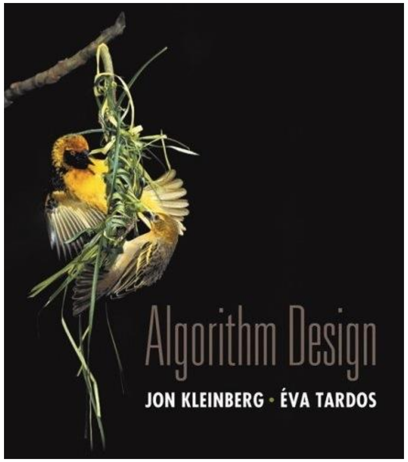
You will get at least 3.4 if

- Your score is >= 65%

Do not come to the class/exam if you feel sick! (There are videos)

Where to get help?

- Ask questions in the class!
- Read the textbook!
- Edstem: Online discussion forum (To be online).
- Office hours:
- Myself: M 2:30-3:30 in CSE 562



Course textbook (optional)

Office hours:

TA	Office hours (Jan 10 - Mar 11)	Room
Andreea Ghizila andreeag@uw	Mon 11:30-12:30pm	4th Floor breakout
Sally Dong sallyqd@uw	Tue 11am-12noon	4th Floor breakout
Jason Waataja jwaataja@uw	Tue 1-2pm	4th Floor breakout
Jeremy Lin linjer23@uw	Tue 2-3pm	4th Floor breakout
Ashwin Banwari ash1152@uw	Fri 2:30-3:30pm	5th Floor breakout
Swati Padmanabhan pswati@uw	Fri 3:30-4:30pm	5th Floor breakout

Starting next week

CSE 421: Introduction to Algorithms

Winter, 2022

Yin Tat Lee

> **MWF 1:30-2:20,** CSE2 G20
> Office hours 2:30-3:30pm in (CSE 562 or 5th Floor breakout)

The class will be on zoom in the first week.
Zoom link: https://washington.zoom.us/j/94725527396

Links:
- Assignment: To be online
- Discussion: To be online
- Textbook: Algorithm Design by Jon Kleinberg and Eva Tardos
  We will cover chapters 1-8.
- Reference: Introduction to Algorithms by CLRS

Grading Scheme:
- Homework (50%)
- Midterm (20%)
- Final (30%)

Tentative Schedule

Lecture	Topic	Slides	References
Part I: Graphs			
Lec. 1 (01/03/22)	Stable Matchings		
Lec. 2 (01/05/22)	Graphs		
Lec. 3 (01/07/22)	Breadth First Search		

Homeworks: [Grading guidelines]

Submission is due via TBA
- Homework 1 due Wednesday, 12-Jan at 1:30PM.
- Homework 2 due Wednesday, 19-Jan at 1:30PM.
- Homework 3 due Wednesday, 26-Jan at 1:30PM.
- Homework 4 due Wednesday, 2-Feb at 1:30PM.
- Homework 5 due Wednesday, 16-Feb at 1:30PM.
- Homework 6 due Wednesday, 23-Feb at 1:30PM.
- Homework 7 due Wednesday, 2-Mar at 1:30PM.
- Homework 8 due Wednesday, 9-Mar at 1:30PM.

Website: cs.washington.edu/421

Stable Matching Problem

Given n men and n women, find a "stable matching".

- We know the preference of all people.

	1st (favorite)	2nd	3rd (least favorite)
Xavier	Amy	Brenda	Claire
Yuri	Brenda	Amy	Claire
Zoran	Amy	Brenda	Claire

Men's Preference Profile

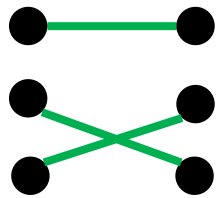
	1st (favorite)	2nd	3rd (least favorite)
Amy	Yuri	Xavier	Zoran
Brenda	Xavier	Yuri	Zoran
Claire	Xavier	Yuri	Zoran

Women's Preference Profile

Stable Matching

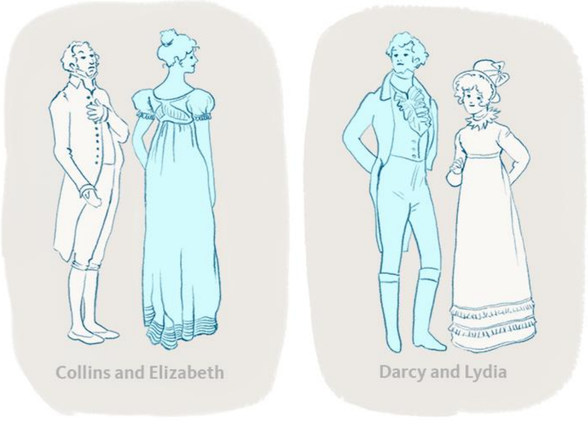
Perfect matching:

- Each man gets exactly one woman.
- Each woman gets exactly one man.



Stability: no incentive to exchange

- an unmatched pair **m-w** is unstable
  if **m** and **w** prefer each other to their current partners.

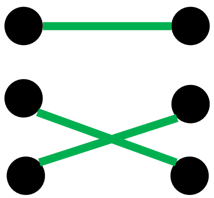


An unstable match:
ELIZABETH AND DARCY LIKE EACH OTHER BETTER THAN THEIR PARTNERS

Stable Matching

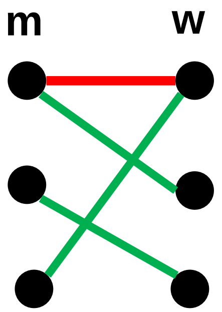
Perfect matching:

- Each man gets exactly one woman.
- Each woman gets exactly one man.



Stability: no incentive to exchange

- an unmatched pair **m-w** is unstable
  if **m** and **w** prefer each other to their current partners.



Stable matching: perfect matching with no unstable pairs.

Stable matching problem: Given the preference lists of n men and n women, find a stable matching if one exists.

Example

Question. Is assignment X-C, Y-B, Z-A stable?

favorite ↓ … least favorite ↓

	1st	2nd	3rd
Xavier	Amy	Brenda	Claire
Yuri	Brenda	Amy	Claire
Zoran	Amy	Brenda	Claire

Men's Preference Profile

favorite ↓ … least favorite ↓

	1st	2nd	3rd
Amy	Yuri	Xavier	Zoran
Brenda	Xavier	Yuri	Zoran
Claire	Xavier	Yuri	Zoran

Women's Preference Profile

Example

Question. Is assignment X-C, Y-B, Z-A stable?

Answer. No. Look at Brenda and Xavier.

favorite ↓ … least favorite ↓

	1st	2nd	3rd
Xavier	Amy	Brenda	Claire
Yuri	Brenda	Amy	Claire
Zoran	Amy	Brenda	Claire

Men's Preference Profile

favorite ↓ … least favorite ↓

	1st	2nd	3rd
Amy	Yuri	Xavier	Zoran
Brenda	Xavier	Yuri	Zoran
Claire	Xavier	Yuri	Zoran

Women's Preference Profile

Example (cont'd)

Question: Is assignment X-A, Y-B, Z-C stable?

Answer: Yes. (X, Y are happy. No one want Z.)

favorite ↓ (1st) … least favorite ↓ (3rd)

	1st	2nd	3rd
Xavier	Amy	Brenda	Claire
Yuri	Brenda	Amy	Claire
Zoran	Amy	Brenda	Claire

Men's Preference Profile

favorite ↓ (1st) … least favorite ↓ (3rd)

	1st	2nd	3rd
Amy	Yuri	Xavier	Zoran
Brenda	Xavier	Yuri	Zoran
Claire	Xavier	Yuri	Zoran

Women's Preference Profile

Existence of Stable Matchings

Question. Do stable matchings always exist?

Answer. Seems unclear in real-world.
Yet, it always exists!

Question. How to find stable matchings?

Propose-And-Reject Algorithm [Gale-Shapley'62]

```
Initialize each person to be free.
while (some man is free) {
    Choose such a man m
    w = 1st woman on m's list to whom m has not yet proposed
    if (w is free)
        assign m and w to be engaged
    else if (w prefers m to her fiancé m')
        assign m and w to be engaged, and m' to be free
    else
        w rejects m
}
```

Switch the pdf for an example.

Basic Properties

```
Initialize each person to be free.
while (some man is free) {
    Choose such a man m
    w = 1st woman on m's list to whom m has not yet proposed
    if (w is free)
        assign m and w to be engaged
    else if (w prefers m to her fiancé m')
        assign m and w to be engaged, and m' to be free
    else
        w rejects m
}
```

- The algorithm ends.
  Is every step valid?
  How many iterations it takes?

- The output is correct.
  It finds a perfect matching that is stable.

How to prove them?

When stuck, try to list out properties of the algorithms.

Observation 1: Men propose to women in decreasing order of preference.

Observation 2: Woman's partner get better and better. Never unmatched once matched.

Summary

- Stable matching problem: Given **n** men and **n** women, and their preferences, find a stable matching.
- Propose-And-Reject algorithm: Guarantees to find a stable matching for any problem instance in $O(n^2)$ steps.
- Steps on solving problems:
  - Formulate the problem
  - Define the algorithm
  - Prove the algorithm ends
  - Prove the output is correct

Why this problem is important?

In 1962, Gale and Shapley published the paper
“College Admissions and the Stability of Marriage”
To
“The American Mathematical Monthly”

COLLEGE ADMISSIONS AND THE STABILITY OF MARRIAGE

D. GALE* AND L. S. SHAPLEY, Brown University and the RAND Corporation

1. Introduction. The problem with which we shall be concerned relates to the following typical situation: A college is considering a set of n applicants of which it can admit a quota of only q. Having evaluated their qualifications, the admissions office must decide which ones to admit. The procedure of offering admission only to the q best-qualified applicants will not generally be satisfactory, for it cannot be assumed that all who are offered admission will accept. Accordingly, in order for a college to receive q acceptances, it will generally have to offer to admit more than q applicants. The problem of determining how many and which ones to admit requires some rather involved guesswork. It may not be known (a) whether a given applicant has also applied elsewhere; if this is known it may not be known (b) how he ranks the colleges to which he has applied; even if this is known it will not be known (c) which of the other colleges will offer to admit him. A result of all this uncertainty is that colleges can expect only that the entering class will come reasonably close in numbers to the desired quota, and be reasonably close to the attainable optimum in quality.

David Gale (1921-2008)
PROFESSOR, UC BERKELEY

Lloyd Shapley
PROFESSOR EMERITUS, UCLA

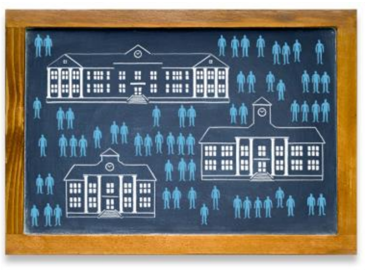
Why this problem is important?


Alvin Roth
PROFESSOR, STANFORD

Alvin Roth modified the Gale-Shapley algorithm and apply it to

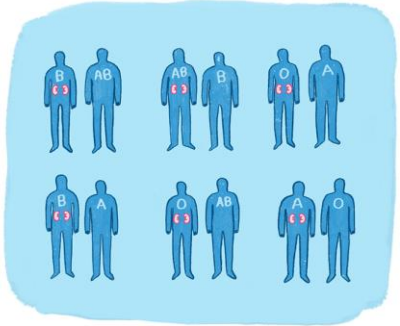
- National Residency Match Program (NRMP), a system that assigns new doctors to hospitals around the country. (90s)


- Public high school assignment process (00s)



- Helping transplant patients find a match (2004)

(Saved >1,000 people every year!)



Blood types: A, B, AB and O

Reference: https://medium.com/@UofCalifornia/how-a-matchmaking-algorithm-saved-lives-2a65ac448698

Why this problem is important?

Some of the problems in this course may seem obscure.

But their abstraction allows for variety of applications.

Shapley and Roth got the Nobel Prize (Economic) in 2012.
(David Gale passed away in 2008.)