1. Administratives

Website: cs.nyu.edu/fall24/CSCI-UA.0310-007/index.html

Lecture: Tue Thu 9:30-10:45 AM, WWH 312

Recitation: Fri 9:30-10:45 AM, Bobst LL138

Office Hours: Jiaxin Guan, Tue 2-4 PM, WWH 412
Jiaming Li, Wed 3-4 PM, WWH412

Grade: 30% HW, 30% Midterm, 40% Final
Overall grade is curved: ~1/3 A, 1/3 B, 1/3 C
No curving down! i.e. ≥90 guarantees A⁻, ...

Homework: ~weekly due Wed 10PM, 11 in total, drop lowest two
No further extensions/accomodations given
Typeset in LaTeX, submit on Gradescope

Collaboration: Encouraged to discuss with classmates, but must
① List discussion partners, and
② Write up your own solution
✗ Look at other student's solution writeup
✗ Show your written solution to others
✗ Look for answers online

Violations will be penalized and reported to the dept.!!!

Midterm: Thu 10/24 In Class 1 double-sided page of note

Final: Thu 12/19 8:00-9:50 AM 2 double-sided pages of note

Participation (lecture, campuswire etc.): 0%, but factored in for borderlines

Textbook: Introduction to Algorithms by Cormen, Leiserson, Rivest & Stein [CLRS]

Questions: Campuswire   campuswire.com/p/GB425CFFD
Email   code: 2692

2. Course Overview

Algorithms: procedures for solving problems

What problems?
- Sorting
- Fast multiplication
- Data structures
- Graph problems

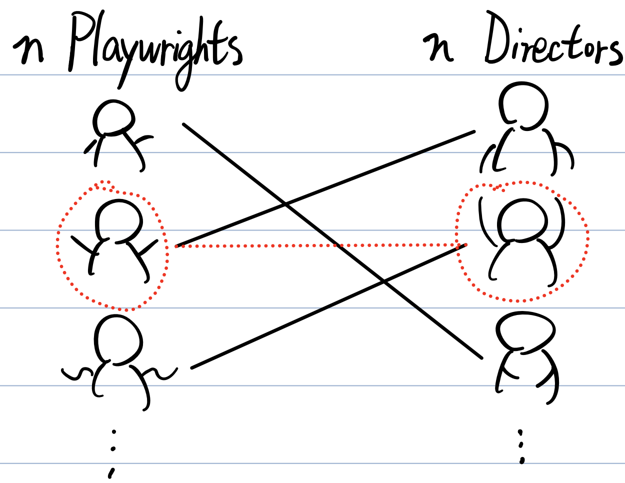
Learn general techniques:
- Divide and Conquer
- Greedy
- Dynamic Programming

Overall Goals:
- Essential analytic skills
- Abstractions and algorithms for problems
- Communication & Proof

(- tech interviews? Shhh~)

3. Stable Matching

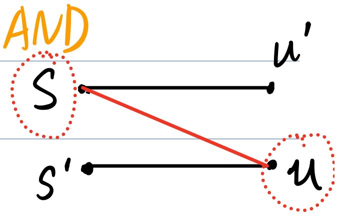
Why study algorithms? Solve real world problems!



Stable Matching (Marriage) Problem:

- Match n students to n schools (1-to-1)
- Preferences: Each student ranks all schools
  Each school ranks all students

  e.g. $A: X > Y > Z$
  $Y: A > C > B$
- Unstable pair: unmatched (s, u), but
  - ▷ s prefers u over current school, AND
  - ▷ u prefers s over current student



- Stable: $\nexists$ unstable pairs

<u>e.g.</u>

Students	Schools
A : $Y > X > Z$	X : $A > B > C$
B : $X > Y > Z$	Y : $B > C > A$
C : $X > Y > Z$	Z : $C > A > B$

A-Z, B-Y, C-X Stable?

No. B-X is unstable pair.

A-X, B-Y, C-Z Stable?

Yes! Each school gets its fav. student.

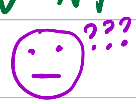
Formalizing Stable Matching:

- Bipartite Graph:

  Graph with V broken into two parts,
  only have edges across two parts

- Matching: Set of edges (max degree 1)
- Perfect Matching: all vertices have degree 1
- "Stable": as defined above

Perfect Matching? ✓ iff # of students & schools the same

Stable Perfect Matching? 

Brute Force:

$n!$ of those

Loop through all possible perfect matchings:

    check if stable

check all $n(n-1)$ pairs

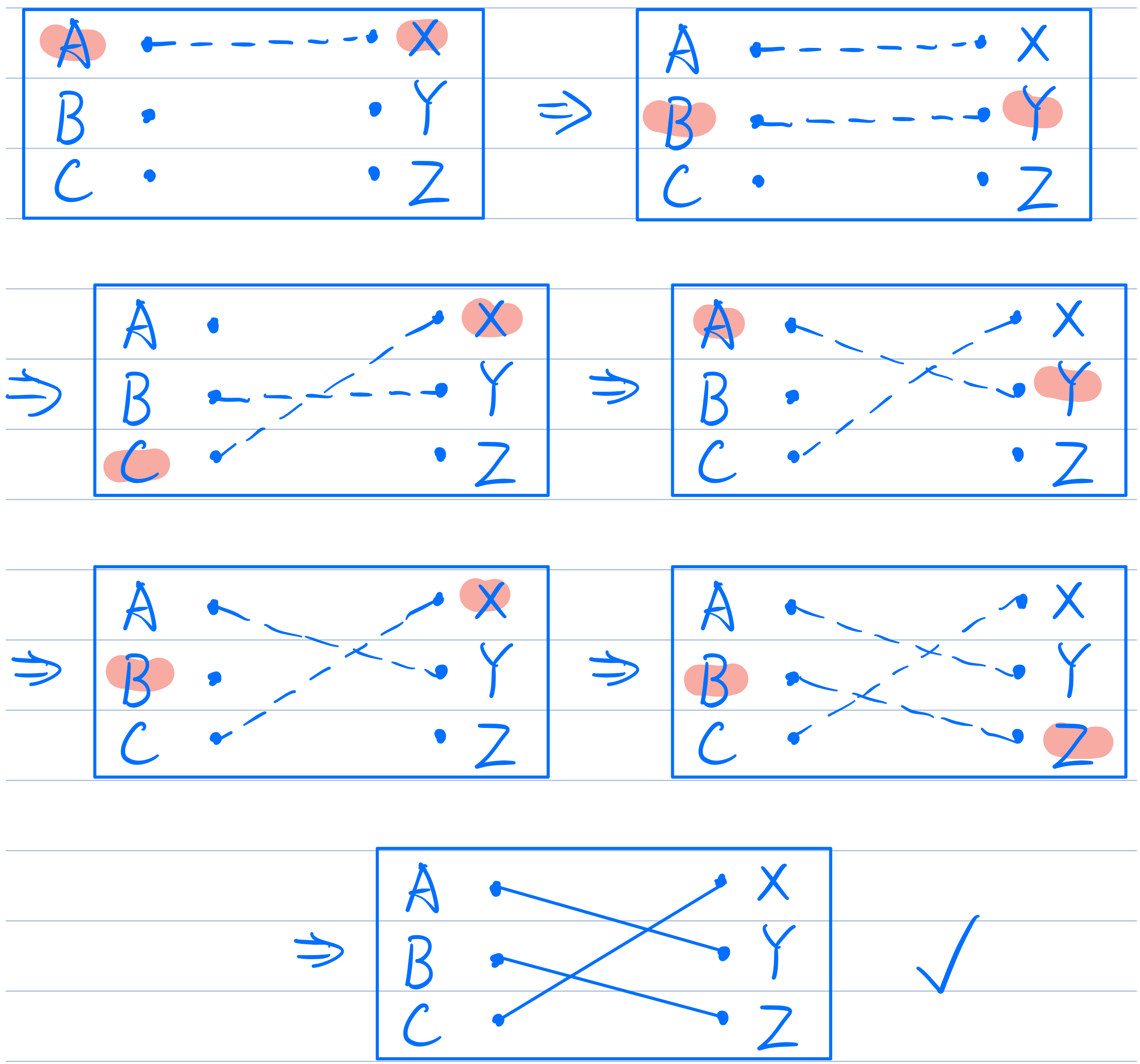
Total: $n! \cdot n(n-1)$ ☹

Gale-Shapley Algorithm (1962):

```
Set all students and schools to be "free"
While (∃ free student):
    Choose such a free student s
    u = first school on s's list that s
        hasn't applied to yet
    If (u is free):
        pair s and u and mark as "matched"
    Else If (u prefers s to u's current match s'):
        pair s and u and mark as "matched"
        mark s' as free
    EndIf
EndWhile
```

e.g.

Students	Schools
A: $X > Y > Z$	X: $C > A > B$
B: $Y > X > Z$	Y: $A > B > C$
C: $X > Z > Y$	Z: $A > B > C$



Claims:

① Once a school gets matched, remains matched
Sequence of its students improves ↑

② The sequence of schools that a student applies to gets worse and worse ↓

③ If at some point s is free, then s hasn't applied to all schools yet

Proof: Assume s has applied to all schools, so all schools are matched (by ①), therefore all students are matched (b/c # equal)
✗

Proof of Correctness

Proof: – The algorithm always terminates w/ a perfect matching (from ③)

- Now we show this matching is stable:
Assume towards contradiction that it's not.
i.e. $\exists$ an unstable pair (s, u) s.t.
(a) s prefers u to its current match u'
(b) u prefers s to its current match s'
Notice that s's last application was to u'

Q: has s applied to u yet?

▷ No ⇒ s prefers u' to u (by ②)
Contradiction w/ (a)!

▷ Yes ⇒ u rejected s (rightaway or later)
⇒ u prefers s' to s (by ①)
Contradiction w/ (b)!

Contradiction in both cases!
⇒ Matching is stable. ⋈

Cor: Stable matching always exists.

Runtime Analysis:

- Total # of possible application?
  $\#(s, u)$ pairs $= n^2$
- Each iteration ⟺ one application
  (iteration: constant time)
- Total running time:
  ☆ Worst Case: "On the order of" n^2
  Best Case: "On the order of" n

By default, we talk about worst case runtime

There's also average case runtime, but analyzing the average case runtime for Gale-Shapley is out of the scope of this course.

(it's "on the order of" $n \log n$ if you're curious)

We will see how to formalize "on the order of" next lecture ...

Whenever you give an algorithm, you should argue:

1. Correctness
2. Runtime Analysis