Lecture 4

CSE 331

Feb 8, 2021

Please do keep on asking Qs!

The only bad question is the one that is not asked!

Not just technical Qs but also on how the class is run

HW 0 solutions are posted

- And go over incorrect proofs

Read the syllabus and hw policy CAREFULLY!

Syllabus Quiz

Options

- View handin history
- View writeup
- Download handout

Due: February 28th 2021, 3:47 pm

Last day to handin: February 28th 2021, 5:47 pm

No graded material will be handed back till you pass the syllabus quiz!

236 (out of 275) already completed!

Academic Integrity

Question 1: Sharing my answers to this syllabus quiz with other 331 students

- ○ Is OK if I do it to help out a friend
- ○ It does not matter since there is no grade attached with it
- ○ Is an academic integrity violation and should not be done
- ○ Is an academic integrity violation but I can take the chance

Question 2: Penalty for academic violation in CSE 331 is an automatic

- ○ Warning and a chance to make-up
- ○ A zero in the assignment AND a letter grade reduction (for first violation across all CSE courses) and an F in the course (for 2nd violation across all CSE courses)
- ○ A zero in the corresponding assignment and nothing else
- ○ Expulsion from UB

Separate Proof idea/proof details

</> Note

Notice how the solution below is divided into proof idea and proof details part. **THIS IS IMPORTANT: IF YOU DO NOT PRESENT A PROOF IDEA, YOU WILL NOT GET ANY CREDIT EVEN IF YOUR PROOF DETAILS ARE CORRECT.**

Proof Idea

As the hint suggests there are two ways of solving this problem. (I'm presenting both the solutions but of course you only need to present one.)

We begin with the approach of reducing the given problem to a problem you have seen earlier. Build the following complete binary tree: every internal node in the tree represents a "parent" RapidGrower while its two children are the two RapidGrowers it divides itself into. After s seconds this tree will have height s and the number of RapidGrowers in the container after s seconds is the number of leaf nodes these complete binary tree has, which we know is 2^s. Hence, the claim is correct.

The proof by induction might be somewhat simpler for this problem if you are not comfortable with reduction. In this case let $R(s)$ be the number of RapidGrowers after s seconds. Then we use induction to prove that $R(s) = 2^s$ while using the fact that $2 \cdot 2^s = 2^{s+1}$.

Proof Details

We first present the reduction based proof. Consider the complete binary tree with height s and call it $T(s)$. Further, note that one can construct $T(s+1)$ from $T(s)$ by attaching two children nodes to all the leaves in $T(s)$. Notice that the newly added children are the leaves of $T(s+1)$. Now assign the root of $T(0)$ as the original RapidGrower in the container. Further, for any internal node in $T(s)$ ($s \geq 0$), assign its two children to the two RapidGrowers it divides itself into. Then note that there is a one to one correspondence between the RapidGrowers after s seconds and the leaves of $T(s)$. Then we use the well-known fact (cite your 191/250 book here with the exact place where one can find this fact): $T(s)$ has 2^s leaves, which means that the number of RapidGrowers in the container after s seconds is 2^s, which means that the claim is correct.

TA office hours will start tomorrow (will post today on Piazza)

Matching Employers & Applicants

Input: Set of employers (E)
Set of applicants (A)
Preferences

Output: An assignment of applicants to employers that is "stable"

For every x in A and y in E such that x is **not** assigned to y, either

(i) y prefers *every* accepted applicant to x; or

(ii) x prefers her employer to y

What questions to think about?

What questions to think about?

1) How do we specify preferences? Preference lists

2) Ratio of applicant vs employers 1:1

3) Formally what is an assignment? (perfect) matching

4) Can an employer get assigned > 1 applicant? NO

5) Can an applicant have > 1 job? NO

6) How many employer/applicants in an applicants/employers preferences? All of them

7) Can an employer have 0 assigned applicants? NO

8) Can an applicant have 0 jobs? NO

Lost in Notation….

CSE 331 Spring 2021 Schedule

Previous schedules: 2020, 2019, 2018, 2017, 2016, 2014

Future Lectures

The topics for lectures in the future are tentative and subject to change. Also the links for future lectures are from Spring 2020 and Fall 2019.

Date	Topic	Notes
Mon, Feb 1	Introduction S21 S20 F19	**HW 0 out** Week 1 recitation notes
Wed, Feb 3	Main Steps in Algorithm Design S20 F19	
Fri, Feb 5	Stable Matching Problem S20 F19 x^2	[KT, Sec 1.1]
Mon, Feb 8	Perfect Matchings S20 F19 x^2	**HW 0 due** [KT, Sec 1.1]

notations

Non-feminist reformulation

n men

Each with a preference list

n women

Match/marry them in a “stable” way

Proof Details: Q1(b) on HW0

(incomplete) exam

Argument does not use ANYTHING about the problem statement!

Follows from part (a)

...of perfect matchings with n men and n wo...n

This assumes number of perfect matchings only depends on n

Base case: $P(1) = 1! = 1$

Inductive hypothesis: Assume that $P(n-1) = (n-1)!$

Inductive step: Note that $P(n) = n*P(n-1) = n*(n-1)! = n!$

What are the issues with the above “proof”?

Proof Details: Q1(b) on HW0

Incorrect (incomplete

Needs justification

Claim 1: Number of perfect matchings is = number of permutations of 1…n

Claim 2: Number of permutations of 1…n is n!

Needs justification

Claims 1 + 2 prove the result

Follow from 191 (?)

What are the issues with the above proof?

Proof by contradiction for Q1(a)

Incorrect example

Assume for contradiction there is an example where number of perfect matchings depends on the identities of the men and women.

You can only assume things about the example directly implied by it being a counter-example

Let $n = 1$ and consider two cases

(1) M = {BP} and W = {JA}

(2) M = {BBT} and W = {AJ}

In both cases the number of perfect matchings is $1 = 1!$

Hence contradiction.

There is NO contradiction

What are the issues with the above proof?

Questions/Comments?

Matching and Perfect matching

A matching $S \subseteq M \times W$ such that following conditions hold:

S is a **set** of pairs (m,w) where m in M and w in W

exactly

(1) For every woman w in W, exist *at most* one m such that (m,w) in S

exactly

(2) For every man m in M, exist *at most* one w such that (m,w) in S

Perfect matching

On matchings

Michael

Pam

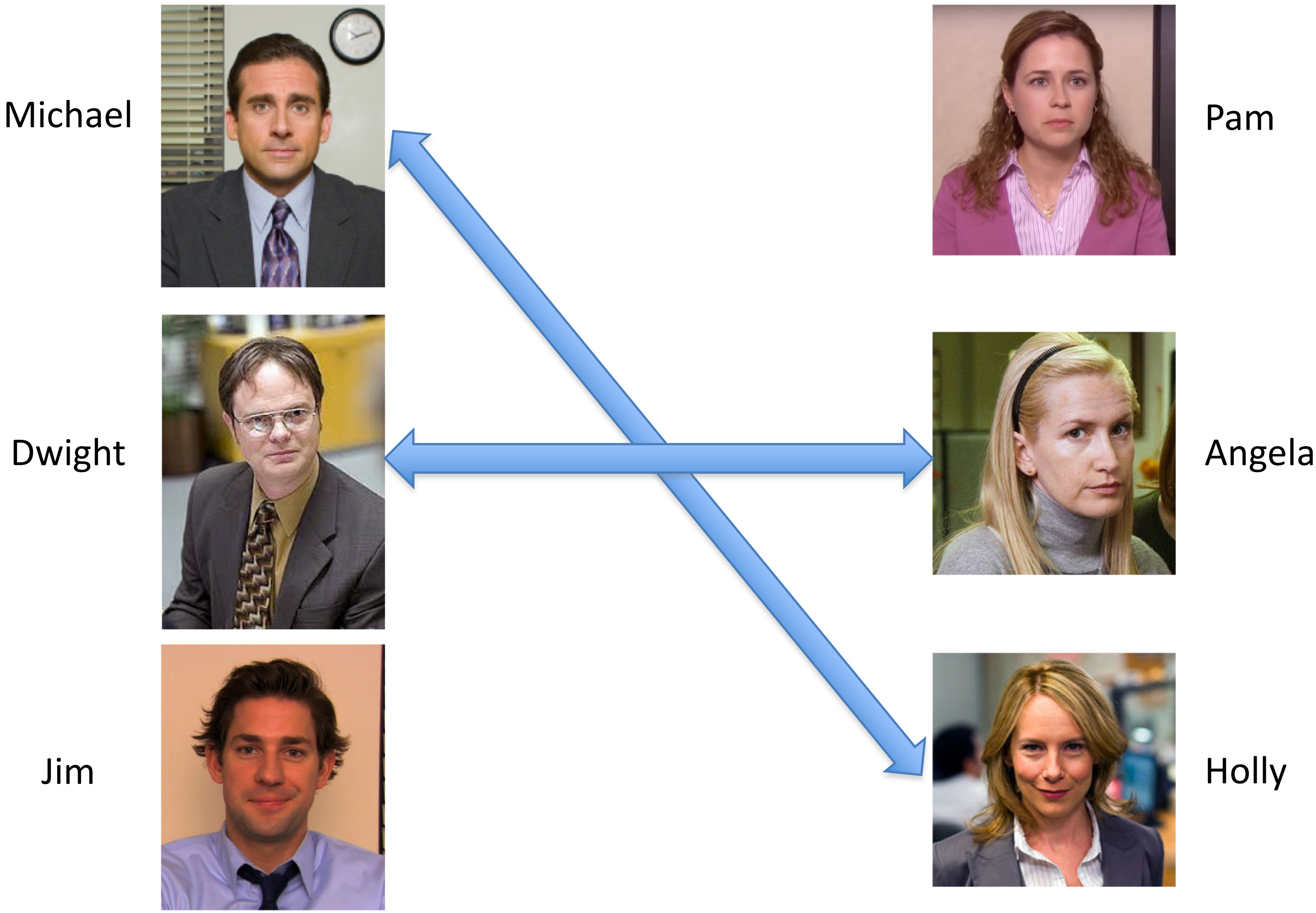
Dwight

Angela

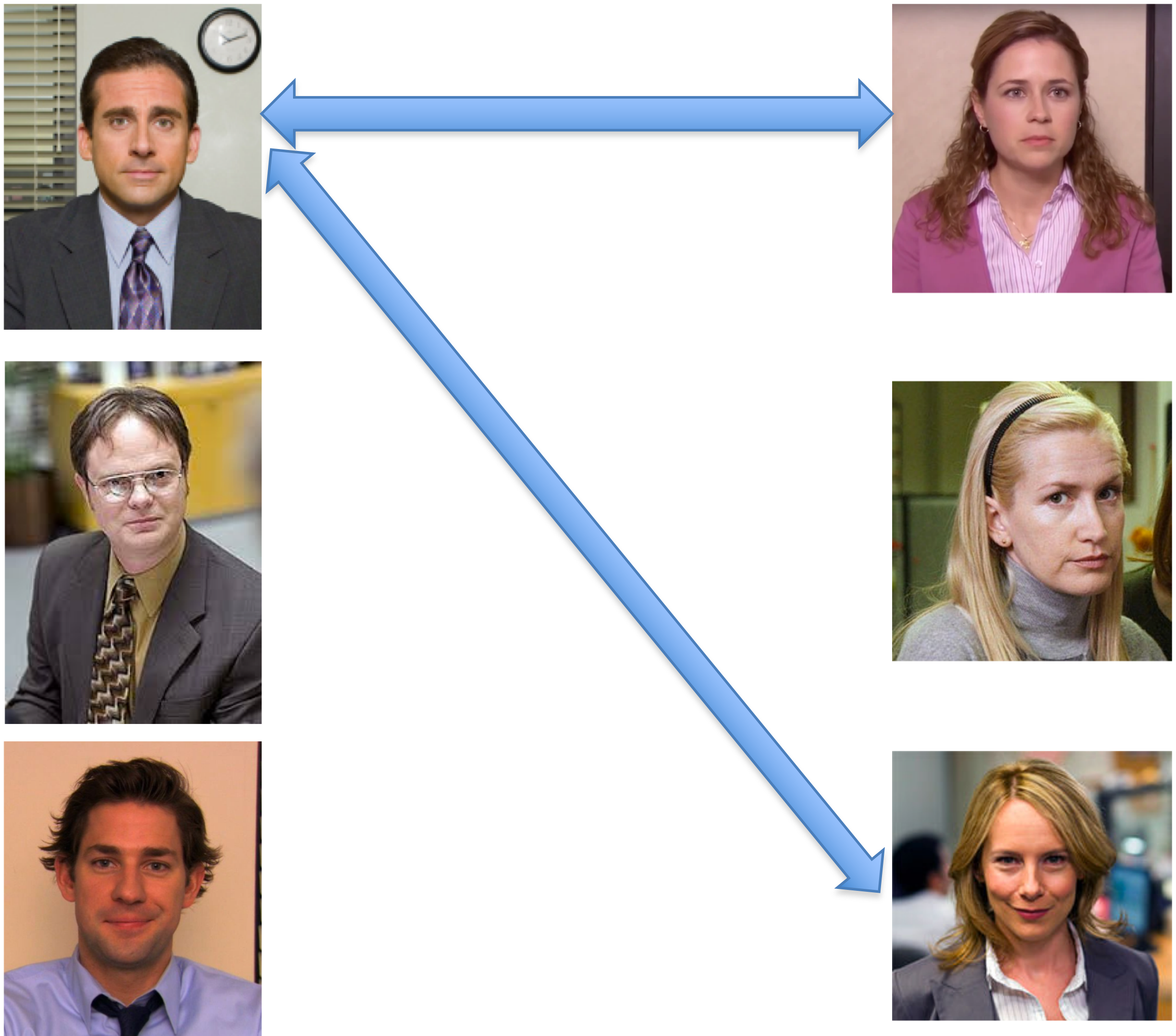
Jim

Holly

A valid matching



Not a matching



Michael

Dwight

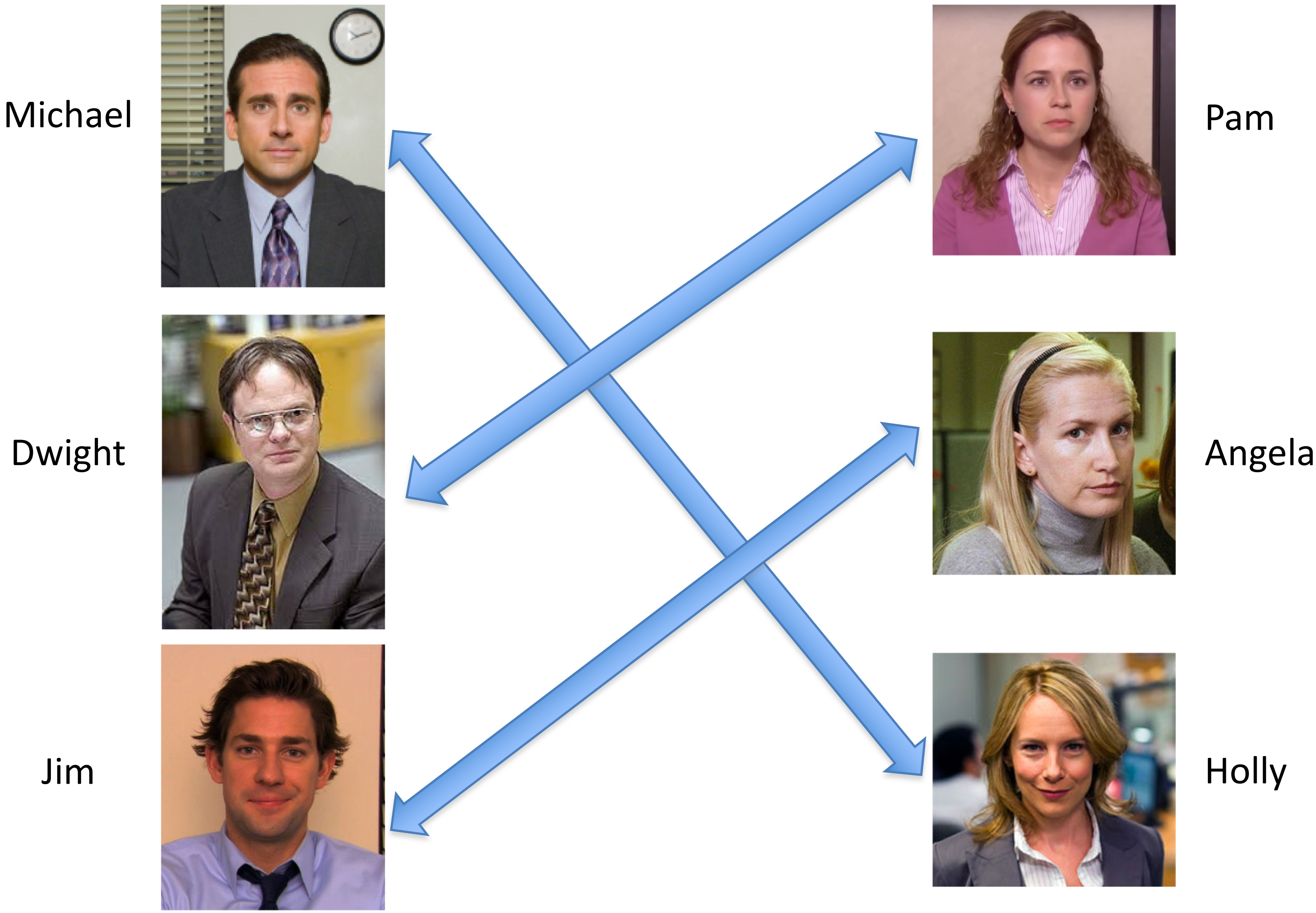
Jim

Pam

Angela

Holly

Perfect matching



Back to couple more definitions

Preferences

Instability