COMPSCI 311: Introduction to Algorithms **Fall 2022**

Discussion 3

Your Name: ____________________ Collaborators: ____

You will be randomly assigned groups to work on these problems in discussion section.

Problem 1. Proofs. Read the following claim and proof, adapted from Solved Exercise 1 in Chapter 1.

Consider an instance of the stable matching problem where all colleges and students are divided into two categories: *good* and *bad*. For some number k between 1 and n there are exactly k *good* colleges and exactly k *good* students. The other $n-k$ students and $n-k$ colleges are *bad*. Every student prefers all good colleges to all bad colleges, and every college prefers all good students to all bad students.

Claim. In every stable matching, every good college is matched to a good student.

Proof:

- Let's try to prove this by contradiction.
- Suppose M is a stable matching in which a good college c is matched to a bad student s.
- What do the other pairs in M look like?
- Could it be the case that every good college is matched with a good student in M? No,
- We know there are k good colleges and k good students.
- Because c is matched to a bad student, there are at most $k-1$ good colleges remaining, and at least one good student must be matched to a bad college.
- Call this student s', and consider the pair (c, s').
- This pair is not matched in M, but both are *good*, so each prefer the other to whomever they are matched with.
- Thus, (c, s') is an instability.
- This contradicts our assumption that M is stable.
- Therefore, in every stable matching, every good college is matched to a good student.

Every proof starts with some **premises**, which are facts that are either given or assumed. The proof consists of a **sequence of statements** that argue to a **conclusion** (what you are trying to prove). Most statements in proofs do one of the following:

1. State a **premise**
2. Declare some **notation**
3. State a new fact that can be deduced via straightforward logic from the previous statements (**deduce**)
4. Provide commentary or intuition (**comment**)
5. State the **conclusion**

Label each statement in the proof with the best match from this list. (There can be some ambiguity if one sentence does multiple things or is part of a certain logical construct such as the setup of a proof by contradiction. Say "ambiguous" if you think it is is the case.)

Try reading the proof without any of the "comment" statements. Does it sound OK? The textbook tends to interleave a lot of comments into problems and proofs. These are supplementary and sometimes distracting. I recommend using them somewhat sparingly, and focusing on making all of the other statements very precise.

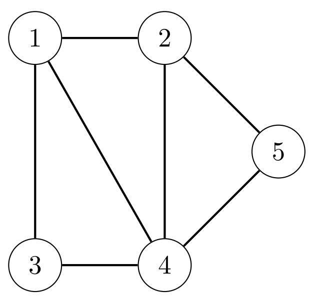
Problem 2. Graphs, proofs. The degree of node v in a graph is the number of edges incident to v (equivalently, the number of neighbors of v). One problem in proofs is statements that are not precise enough. Consider the claim and proof below. Identify at least two statements or phrases in the proof that are imprecise or not fully justified.

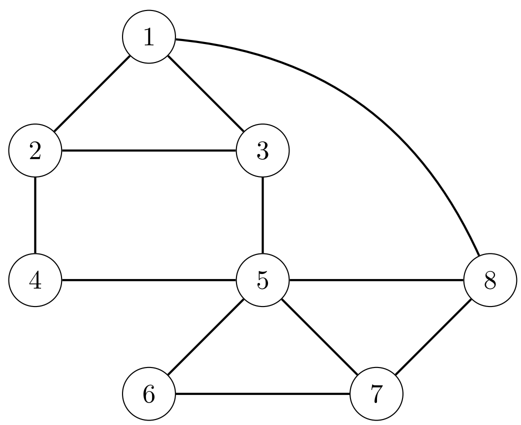
Claim: Let G be an undirected graph where every node has degree two or more. Then G has a cycle.

Proof: Since every node has two or more neighbors, we can construct a path starting from any node and keep going on that path without ever turning around. The path must eventually loop back on itself, so there is a cycle.

Now discuss with your group how you could make this more precise.

Problem 3. Traversal Examples. Draw the BFS tree and DFS tree for each of the example graphs below.





Problem 4. K&T Ch 3, Ex 7. Claim: Let G be a graph on n nodes, where n is an even number. If every node of G has degree at least $n/2$, then G is connected. Decide whether you think the claim is true or false, and either give a proof of the claim or give a counterexample.

Problem 5. Puzzle (for fun)

(a) You are to cut out some pieces of paper. You must be able to place your pieces of paper on an 8×8 checkerboard so that they exactly cover the checkerboard, minus -any- one square. That is, once your pieces are cut, if I identify to you -any- of the squares on the checkerboard, you must be able to arrange all your pieces of paper (flat, not folded) on the checkerboard so that:

   (i) the pieces of paper don't overlap

   (ii) the pieces of paper don't cover any area outside the checkerboard

   (iii) the pieces of paper cover all of the checkerboard except the square I chose

   The problem: figure out how to do this with three pieces of paper.

   For example, it is easy to see how to do this with 63 pieces of paper – use 63 1x1 squares of paper. Or, 32 pieces — use a 4x8 piece and 31 1x1 pieces...

(b) Relate your solution to a specific Divide-and-Conquer algorithm you learned prior to CS 311. (Hint: what if the board were 8x1 instead of 8x8? What if it were 16x16?)