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

**Problem 1.** Running Time. Each of the algorithms below prints some number of “X” characters and “Y” characters. For each algorithm, unless specific instructions are given, do the following: first write the output that would be produced with \( n = 4 \), then give a tight bound on its running time using Big-O notation.

1. Algorithm Print1(\( n \))
   
   \[
   \text{for } i = 1 \text{ to } n \text{ do} \\
   \text{print "X"} \\
   \text{for } j = 1 \text{ to } n \text{ do} \\
   \text{print "Y"}
   \]

2. Algorithm Print2(\( n \))
   
   \[
   \text{for } i = 1 \text{ to } n \text{ do} \\
   \text{print "X"} \\
   \text{if } i == 1 \text{ then} \\
   \text{for } j = 1 \text{ to } n \text{ do} \\
   \text{print "Y"}
   \]

3. Algorithm Print3(\( n \))
   
   \[
   \text{for } i = 1 \text{ to } n \text{ do} \\
   \text{print "X"} \\
   \text{if } i \leq n/2 \text{ then} \\
   \text{for } j = 1 \text{ to } n \text{ do} \\
   \text{print "Y"}
   \]

4. A poem has \( m \) words printed on \( n \) lines. There may be any number of blank lines. Consider the following algorithm:

   Algorithm PrintPoem(\( \ldots \))
   
   \[
   \text{for each line of the poem do} \\
   \text{print "X"} \\
   \text{for each word on the line do} \\
   \text{print "Y"}
   \]

   • How many times is “X” printed?
   • How many times is “Y” printed?
   • Give a Big-O running time bound for the algorithm in terms of \( m \) and \( n \).
   • Now assume there are no blank lines. Can you give a Big-O bound that depends on only one of \( m \) or \( n \)?

5. Algorithm Print4(\( n \))
   
   \[
   \text{while } n > 1 \text{ do} \\
   \text{print "X"} \\
   \text{Set } n = n/2
   \]

   (Use \( n = 16 \) instead of \( n = 4 \) for this problem)
Problem 2. 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 3. **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 4.** Traversal Examples. Draw the BFS tree and DFS tree for each of the example graphs below.