Berkeley Math Circle, Intermediate I, 10/23/2024

Warm-up activity (groups of 2). You have red and blue pieces of paper. The verifier should imagine they are colorblind (if they aren't). The prover's job is to try to convince the Verifier that the two pieces of paper are different even though they look identical to them. Take turns playing the roles of Prover and Verifier.

Solving problems with constraints

I have these ingredients: peanut butter, ground beef, tofu, egg, ketchup, apples. I am planning up to 3 meals: breakfast, lunch and dinner, and I want to use all the ingredients. Let's say the following pairs **should not** be in a meal together:

- Peanut butter and ground beef
- Peanut butter and tofu
- Peanut butter and egg
- Peanut butter and ketchup
- Ground beef and apples
- Ground beef and tofu
- Tofu and apples
- Tofu and ketchup
- Egg and apples
- Ketchup and apples

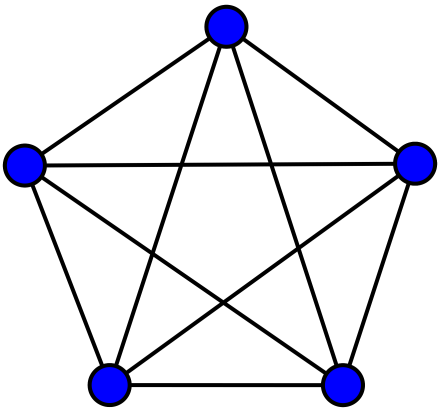
Problem: Try to figure out a way to make 3 meals that use all the ingredients.

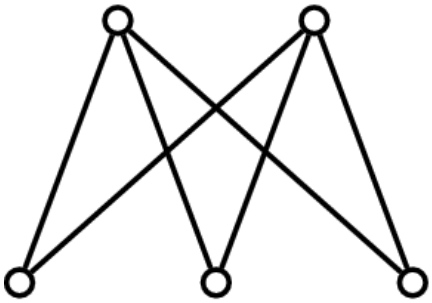
Graph coloring

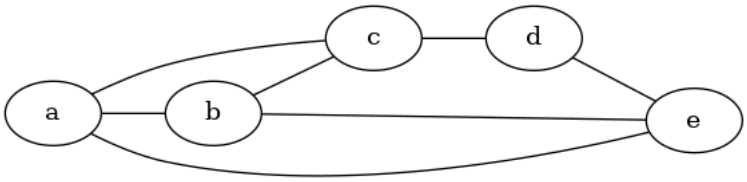
We can look at a graph G and ask if it's possible to put one of k colors on each vertex so that every edge has different colored endpoints. In other words, every vertex should have a different color from its neighbors.

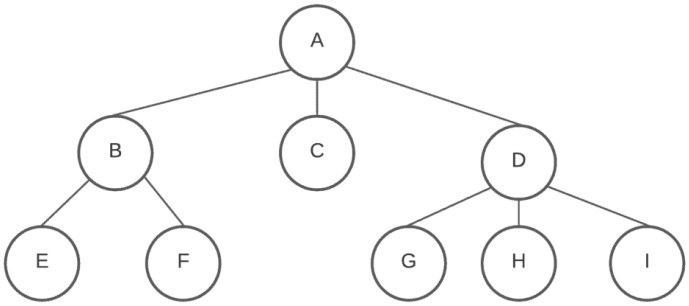
If it is possible, we say the graph is **k-colorable** and we call an assignment of colors to vertices a **k-coloring.**

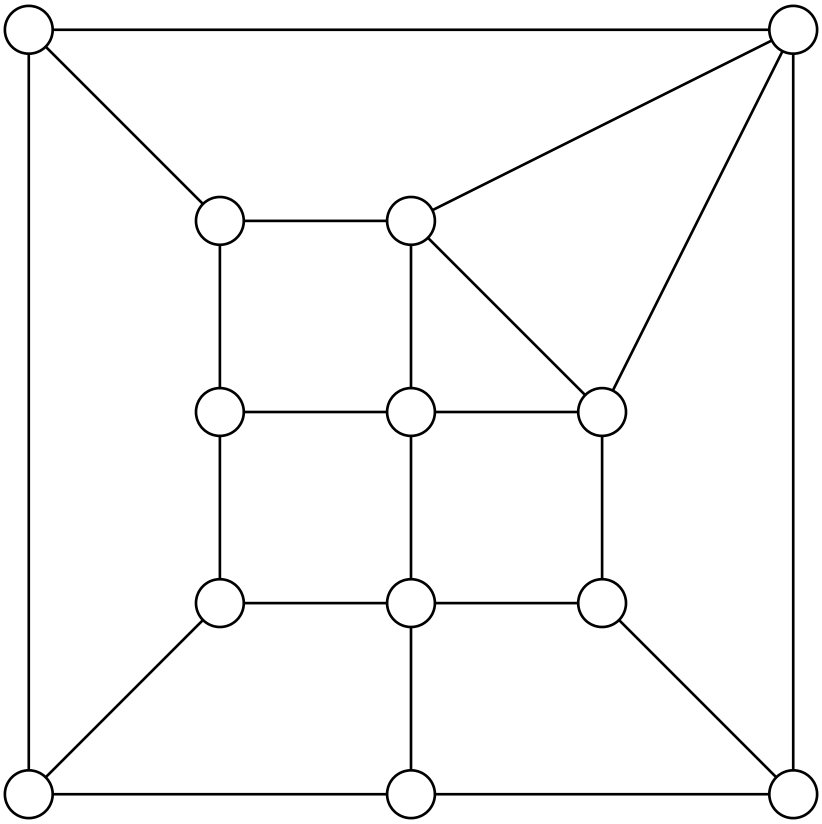
Problem: How many colors do you need to color these graphs? If you don't have colored pens just write R, G, B, Y, P for red/green/blue/yellow/purple.











Proving colorings

Problem: Why is there a chance that the Verifier will catch a lying prover?

Problem: Say there are m edges. What is the probability the Verifier will catch a lying prover?

We know doing this once will allow the Verifier to catch the prover with some probability, so we have them repeat it many times. Lying once, they may get away with, but if they try to lie many times, eventually they will get caught.

Problem: If we repeat k times, what is the probability that a lying prover will be caught at least once?

Problem: Say we are running the protocol with a graph of size 100. How many times do we need to run the protocol for the chance of a cheating prover winning to be less than 1%?

We say that this protocol is zero-knowledge. That means the verifier learns nothing from the interaction except that a 3-coloring exists. Whatever the verifier sees during the interaction, they could have come up with on their own.

Problem: Why is it that this protocol is zero knowledge? What does the verifier see during their conversation with the prover? Could they generate that on their own?

Reviewing

- Explain in your own words to a classmate what zero knowledge proofs are.
- Can you think of anything in your own life that it would be interesting to use zero knowlege proofs for?
- Are these really proofs? How do they compare to the usual notion of proof that you have seen?
- What kind of problems have short proofs? E.g., we have a short proof for graphs being 3-colorable, but we don't seem to have one that a graph **is not** 3-colorable.
- What kind of problems do you think have short proofs that are also zero-knowledge?

For fun (if you know what a tree is): Prove that every tree is 2-colorable.
Challenge: Suppose G is a graph with n vertices that needs at least k colors to color. Prove that G has at least $\binom{k}{2}$ edges.