Problem 1 (Formation)

Before you begin the group activity for this week, identify:

- A *Scribe*: who will be responsible for creating a shared document and turning in your solutions for this week's problems.
- A *Timekeeper*: who will keep track of the timer for weekly activities that require timekeeping.
- A *Driver*: who will be responsible for asking people to share or otherwise kickstarting conversation if group chat slows down.

For this problem, simply identify your teammate's roles in the space below.

If you cannot attend a given class period, please work with your group on how you will contribute to the group's work either asynchronously (*e.g.*, by doing the work yourself) or outside of the class period (*e.g.*, by chatting or meeting outside of class). If there are any issues in coordinating work in your group, please let me know as soon as possible!

Problem 2 (Tours, Approximately)

A polynomial-time approximation scheme (PTAS) is an approximation algorithm for a minimization problem that produces solutions that are within a factor of $(1+\epsilon)$ of the optimal solution, for some constant parameter ϵ. In the book, Sipser describes a 2-approximation algorithm for VERTEX-COVER which is a PTAS with $\epsilon = 1$. Several NP-complete problems has PTAS.

Recall that a *Hamiltonian cycle* is a cycle in a graph that touches every vertex at least once. From this, we can discuss the *traveling salesman problem* (TSP) which requires us to find the shortest Hamiltonian cycle in a weighted undirected graph. For simplicity's sake, when we talk about TSP, we assume that the graph is complete, *i.e.*, every location is reachable from every other location. It turns out that it is provable that TSP has no PTAS which is not good since TSP occurs in many real-life situations!

In order to gain a tractable approximation for TSP, we need to add in an extra constraint, *the triangle inequality*, which states that for any triple of vertices u, v, w:

$$l(u,v) + l(v,w) \geq l(u,w)$$

where $l(u,v)$ is the weight of the edge uv. If we add this constraint to TSP, then we can obtain a 2-approximation scheme for TSP.

(a) Individually, spend no more than 2 minutes making sure you understand what the triangle inequality says. Come up with a small example graph that exhibits the triangle inequality and one that does not. Come back with your group and answer the following question: if we know that the triangle inequality holds of a graph, what do we know about the shortest path between any pair of vertices in the graph?

(b) As a group, spend no more than 3 minutes reviewing the concept of a *minimum spanning tree* (MST) for a graph. In particular, what is a MST and how can you compute one efficiently? You do not need to include anything in your write-up for this part.

(c) Now, spend 10 minutes individually taking this idea of a MST and deriving an approximation algorithm for the TSP. Come back as a group and agree on a final algorithm and describe it in your write-up. (*Hint*: Think about using the MST of the graph as a starting point for the tour? How can the MST be used to visit all the vertices of the graph? From this, you can view the problem as "fixing up" this tour so it is indeed Hamiltonian.)

(d) Finally, spend another 5 minutes individually to think about why this algorithm is a 2-approximation scheme. That is, the result is no larger than two times the optimal tour of the graph. Come back as a group and summarize your reasoning. You do not need to formally prove this fact, but instead try to give intuition why the factor is specifically 2 with respect to the triangle inequality and MSTs.

Problem 3 (Forthcoming!)

Problem 4 (Forthcoming!)