Precept Outline

- Review of Lectures 23 and 24:
  - Intractability
  - Algorithm Design

A. Review: Intractability

Your preceptor will briefly review key points of this week's lectures.

B. P, NP and NP-completeness

Suppose that problem X is in NP and problem Y is NP-complete. Which of the following can you infer?

1. ( ) X is not NP-complete.
2. ( ) If Y can be solved in polynomial time, then so can X.
3. ( ) If X can be solved in polynomial time, then P = NP.
4. ( ) Y is in P.
5. ( ) If P = NP, then X can be solved in polynomial time.
6. ( ) Y is not in P.
7. ( ) Factoring polynomial-time reduces to Y.
8. ( ) If P = NP, then Y can be solved in polynomial time.
9. ( ) If P $\neq$ NP, then X cannot be solved in polynomial time.
10. ( ) If X cannot be solved in polynomial time, then P $\neq$ NP.

C. Algorithm Design via Reductions: Princeton MSTs (Fall'23 Final)

Consider the classical minimum spanning tree problem and a variant:

- **Classic-MST**: given a connected edge-weighted graph G, find a spanning tree of G with minimal total weight.
- **Princeton-MST**: given an edge-weighted graph G with each edge colored orange or black, find a spanning tree of G that has minimum total weight among all spanning trees that contain all of the orange edges (or report that no such spanning tree exists).

Design an efficient algorithm to solve the Princeton-MST problem on an edge-weighted and edge-colored graph G. *To do so, you **must** model it as a Classic-MST problem on a closely related edge-weighted graph G'.*

For full credit, your algorithm must run in $O(E \log E)$ time, where V and E are the number of vertices and edges in G, respectively.

D. Jeopardy!

If there is time left, your preceptor will lead a Jeopardy! round with categories from topics of the course. Have fun! (And if you don't know the rules, make sure to ask before starting.)