Note: Your TA may not get to all the problems. This is totally fine, the discussion worksheets are not designed to be finished in an hour. The discussion worksheet is also a resource you can use to practice, reinforce, and build upon concepts discussed in lecture, readings, and the homework.

1 Optimization versus Search

Recall the following definition of the Traveling Salesman Problem, which we will call tsp. We are given a complete graph $G$ of whose edges are weighted and a budget $b$. We want to find a tour (i.e., path) which passes through all the nodes of $G$ and has length $\leq b$, if such a tour exists.

The optimization version of this problem (which we call tsp-opt) asks directly for the shortest tour.

(a) Show that if tsp can be solved in polynomial time, then so can tsp-opt.

(b) Do the reverse of (a), namely, show that if tsp-opt can be solved in polynomial time, then so can tsp.

2 Reliable Network

Reliable Network is the following problem: We are given two $n \times n$ matrices (a cost matrix $d_{ij}$ and a connectivity requirement matrix $r_{ij}$) and also a budget $b$. We want to find a graph $G = (\{1, ..., n\}, E)$ such that the total cost of all edges (i.e. $\sum_{(i,j) \in E} d_{ij}$) is at most $b$ and there are exactly $r_{ij}$ vertex-disjoint paths between any two distinct vertices $i$ and $j$.

Show that Reliable Network is NP-Complete.
3 (★★★★) 2-SAT and Variants

In this problem we will explore the variant of 3-SAT called 2-SAT, where each clause contains at most 2 literals (hereby called a 2-clause).

(a) Show that 2-SAT is in \( \mathbf{P} \). (Hint: Note that the clause \( x \lor y \) is equivalent to the implications \( \neg x \Rightarrow y \) and \( \neg y \Rightarrow x \). If a 2-SAT formula is unsatisfiable, what must be true about the set of all such implications?)

(b) The problem of Max-2-SAT is defined as follows. Let \( C_1, \ldots, C_m \) be a collection of 2-clauses and \( k \) a non-negative integer. We want to determine if there is some assignment which satisfies at least \( k \) clauses.

The problem of Max-Cut is defined as follows. Let \( G \) be an undirected unweighted graph, and \( k \) a non-negative integer. We want to determine if there is some cut with at least \( k \) edges crossing it. Max-Cut is known to be \( \mathbf{NP} \)-complete.

Show that Max-2-SAT is \( \mathbf{NP} \)-complete by reducing from Max-Cut. Prove the correctness of your reduction.