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 Decision vs. Search vs. Optimization

Recall that a vertex cover is a set of vertices in a graph such that every edge is adjacent to at least one vertex in this set.

The following are three formulations of the VERTEX COVER problem:

- As a *decision problem*: Given a graph G, return TRUE if it has a vertex cover of size at most b, and FALSE otherwise.
- As a *search problem*: Given a graph G, find a vertex cover of size at most b (that is, return the actual vertices), or report that none exists.
- As an *optimization problem*: Given a graph G, find a minimum vertex cover.

At first glance, it may seem that search should be harder than decision, and that optimization should be even harder. We will show that if any one can be solved in polynomial time, so can the others.

(a) Suppose you are handed a black box that solves VERTEX COVER (DECISION) in polynomial time. Give an algorithm that solves VERTEX COVER (SEARCH) in polynomial time.

(b) Similarly, suppose we know how to solve VERTEX COVER (SEARCH) in polynomial time. Give an algorithm that solves VERTEX COVER (OPTIMIZATION) in polynomial time.

2 Vertex Cover to Set Cover

In the minimum vertex cover problem, we are given an undirected unweighted graph $G = (V, E)$ and asked to find the smallest vertex cover (defined in the previous problem).

Recall the minimum set cover problem: Given a set U of elements and a collection $S_1, \ldots, S_m$ of subsets of U, find the smallest collection of these sets whose union equals U.

Give an efficient reduction from the minimum vertex cover problem to the minimum set cover problem.

3 Exact 3-SAT

In the 3-SAT problem, we have variables x_i and clauses, where each clause is the OR of **at most** three literals (a literal is a variable or its negation). Our goal is to find an assignment of variables that satisfies all the clauses.

The exact 3-SAT problem is just like the 3-SAT problem, except each clause has **exactly** three **distinct** literals.

Give a reduction from 3-SAT to exact 3-SAT. (Hint: Note that $(x \vee y) \wedge (x \vee \neg y)$ is logically equivalent to x).

4 Cycle Cover

In the cycle cover problem, we have a directed graph G, and our goal is to find a set of directed cycles $C_1, C_2, \ldots C_k$ in G such that every vertex appears in exactly one cycle (a cycle cannot revisit vertices, e.g. $a \rightarrow b \rightarrow a \rightarrow c \rightarrow a$ is not a valid cycle, but $a \rightarrow b \rightarrow c \rightarrow a$ is), or declare none exists.

In the bipartite perfect matching problem, we have a undirected bipartite graph (a graph where the vertices can be split into L, R, and there are no edges between two vertices in L or two vertices in R), and our goal is to find a set of edges in this graph such that every vertex is adjacent to exactly one edge in the set, or declare none exists.

Give a reduction from cycle cover to bipartite perfect matching. (Hint: In a cycle cover, every vertex has one incoming and one outgoing edge.)