Q1. Optimization vs Search

Assume TSP can be solved in polynomial time by algorithm T (input: a matrix of distances and a budget), then the following algorithm can solve TSP-OPT in polynomial time:

T' (input: a matrix of distances)
- Set lower bound of budget as \( N \), the number of cities
- Set upper bound of budget as \( NM \), where \( M \) is the largest weight
- Binary search in the budget range from \( N \) to \( NM \), using T on the input of the same matrix of distances, and start with the budget \( \lfloor (N + NM)/2 \rfloor \). If for the given budget there exists a tour, reset the upper bound to be this current budget and repeat the process; else if there does not exist a tour, reset the lower bound to be the current budget and repeat. We stop when the two bounds meet, and that will be the minimum budget with which a tour exists.
- Return the route with the minimum budget

Q2. Proving NP-Completeness by Generalization

(a) SUBGRAPH ISOMORPHISM:
This is a generalization of CLIQUE problem: given graph \( H \) and determine whether there is a clique (a complete graph) of a given size. If we set the \( G \) in SUBGRAPH ISOMORPHISM problem to be the complete graph of the given size, it will be exactly the CLIQUE problem.

(c) MAX SAT:
This is a generalization of SAT problem: given a CNF and find a truth assignment that satisfies all clauses. If we set the \( g \) in MAX SAT problem to be the number of clauses then it is exactly the SAT problem.

(e) SPARSE SUBGRAPH:
This is a generalization of INDEPENDENT SET problem: given a graph and find a set of \( a \) vertices where there are no edges between any pair of vertices. If we set the \( b \) in SPARSE SUBGRAPH to be zero, it becomes INDEPENDENT SET problem.
Q3. Reductions to Integer Linear Programming

For full credits you need to do at least 3 of the 4 reductions below:

1. From Zero-One Equations

The Zero-One Equations asks for an assignment of 0s and/or 1s for all variables satisfying the given linear inequalities. It can be solved by adding the two equations for each variable $x_i$: $x_i \geq 0$, $x_i \leq 1$, in addition to the existed equations so it becomes an instance of Integer Linear Programming.

2. From Subset Sum:

The Subset Sum problem asks for a subset of given set of integers such that the sum of all numbers in this subset is zero. Assume the set of numbers is $\{a_1, a_2, ..., a_n\}$, then it can be solved if the following linear equalities can be satisfied by integer assignment of all variables:

$$a_1x_1 + a_2x_2 + ... a_nx_n = 0, \quad x_i \geq 0 \text{ and } x_i \leq 1 \text{ for all } x_i$$

This is an instance of Integer Linear Programming.

3. From 3SAT:

The 3SAT is to find a Boolean assignment to all variables of CNF where each clause contains exactly 3 literals. Let us define integer-value variable $y_i$ for each variable $x_i$ so we have the following inequalities:

$$y_i \geq 0 \text{ and } y_i \leq 1 \text{ for all variables } x_i$$

Then for each 3-literal clause add one inequation summing up 3 corresponding $y$-variables: if $x_i$ is not negated in the clause, sum $y_i$; else if $x_i$ is negated, sum $(1 - y_i)$.

For example, for the clause $(x_p \lor x_q \lor x_r)$, the following equation will be added:

$$x_p + x_q + x_r \geq 1$$

And, for the clause $(x_i \lor \neg x_j \lor \neg x_k)$, the following equation will be added:

$$x_i + (1 - x_j) + (1 - x_k) \geq 1$$

Finding an assignment of true/ false to satisfy a given 3SAT CNF is equivalent to finding integer solution to the above Integer Linear Programming problem.

4. From Independent Set:

To solve Independent Set problem of finding $k$ vertices of graph $G = (V, E)$ where $|V| = n \geq k$ there are no edges between any pair of the $k$ vertices, first map each vertex $v_i$ to a variable $x_i$ then find an assignment of integers to satisfy the following inequalities:

$$x_i \geq 0, \text{ for each vertex } v_i \in V$$

$$x_i + x_j \geq 0 \text{ and } x_i + x_j \leq 1, \text{ for each edge } (v_i, v_j) \in E$$

$$x_1 + x_2 + ... x_n = k$$