Pre-tutorial questions

Do you know the basic concepts of this week's lecture content? These questions are only to test yourself. They will not be explicitly discussed in the tutorial, and no solutions will be given to them.

1. Reduction.
   (a) What is a polynomial time reduction?
   (b) Is polynomial time reductions transitive?
   (c) What are the three types of standard reductions?
2. Classes
   (a) What's the definition of the class P?
   (b) What's the definition of the class NP?
   (c) Is it known if P=NP?
   (d) How do you prove that a problem is in NP?
   (e) What's the definition of the class NP-complete?
3. How do one prove that a problem is NP-complete?

Tutorial

Problem 1
The input to the *set cover problem* is a collection of subsets $S_1, S_2, \ldots, S_m$ of some universal set U, and a number t. The problem is to determine if there are t sets $S_{i_1}, S_{i_2}, \ldots, S_{i_t}$ whose union equals U; that is, $\cup_{j=1}^{t} S_{i_j} = U$.

1. Prove that the set cover problem is NP-hard.
2. Argue why the Set Cover problem is more general than the Vertex Cover problem.

Problem 2
Consider a generalization of the interval scheduling problem where requests are not a single interval but a collection of d intervals. The objective is to choose a maximum number of requests that do not create any conflicts. More formally, we have n requests. The ith request has associated intervals $I_1^i, I_2^i, \ldots, I_d^i$. A subset S of requests is feasible if for all $i, j \in S$, and $a, b \in [1, d]$ we have $I_a^i \cap I_b^j = \emptyset$. The objective is to select a maximum size set of feasible requests. Show that this problem is NP-hard.

Problem 3
In the *Degree Δ Spanning Tree problem* we are given a graph $G = (V, E)$ and we have to decide whether there is a spanning tree of G whose maximum degree is at most Δ. (Note that Δ is not part of the input, but rather part of the problem definition.) It is known that the Degree 2 Spanning Tree problem (D2ST) is NP-complete. Using this fact, prove that Degree Δ Spanning Tree is NP-complete for all fixed $\Delta > 2$ (DΔST).

Problem 4

Given a graph $G = (V, E)$, a subset of vertices X and a number k, the Steiner Tree problem is to decide whether there is a set $S \subseteq V$ of size at most k such that $G[X \cup S]$ is connected. Consider the following reduction from 3-SAT to the Steiner Tree problem.

Let $\phi = C_1 \wedge \cdots \wedge C_m$ be a boolean formula over variables $x_1, \ldots, x_n$ such that each clause C_i is the disjunction of three literals. We define a graph $G = (V, E)$ and a target k based on ϕ:

1. For each clause C_i we create a vertex $u_i \in X$; for each variable x_j we create two vertices v_j^T and v_j^F that belong to $V \setminus X$. Finally, we add a dummy node $d \in X$.
2. For each clause C_i, if C_i contains the literal x_j then we create the edge (u_i, v_j^T), while if C_i contains the literal $\neg x_j$ then we create the edge (u_i, v_j^F). Finally, we connect d to every v_j^T and v_j^F.
3. We set the target k to be n.

Prove that the reduction is broken. That is, show a 'Yes' instance being mapped to a 'No' instance or vice versa.

Problem 5

Given a graph $G = (V, E)$, a distinguished subset of vertices $X \subset V$ and a number k, the Steiner Tree problem is to decide whether there is a set $S \subseteq V$ of size at most k such that $G[X \cup S]$ is connected.

Prove that this problem is NP-complete.

Problem 6

Given a directed graph $G = (V, E)$ a feedback set is a set $X \subseteq V$ with the property that $G - X$ is acyclic. The Feedback Set problem asks: Given G and k, does G have a feedback set of size at most k? Prove Feedback Set $\leq_P$ Set Cover.

Problem 7

Decision problems involve answering yes/no questions. For example, the independent set problem is "Given an undirected graph G and a number t, does G have an independent set of size t?". In practice we are usually interested in solving optimization problems. For example, the maximum independent set problem is "Given an undirected graph G, find the largest in independent set in G".

Assuming you have a polynomial time algorithm for the independent set problem, show how to solve the maximum independent set problem.

Problem 8

Prove that CLIQUE is NP-complete by using a reduction from 3-SAT.