DESIGN AND ANALYSIS OF ALGORITHMS
Homework 9

Aman Choudhary
MTech Coursework, CSA 2020
Sr No: 17920

January 4, 2021

1 Problem 1

1.1 Max Clique

- Given a graph $G = (V, E)$, a **clique** is a subset of vertices, $V' \subseteq V$ such that the sub-graph $G' = (V', E)$ induced by V' is complete.
- The objective of this problem is to decide whether there exists a clique in G, such that $|V'| \geq k$, where k is a positive integer.
- The above problem is *NP-Complete* because it belongs to NP and $SAT \leq_p Max\ Clique$.

1.2 3-Coloring

- **Graph Coloring** is the procedure of assigning a color to every vertex of a graph $G = (V, E)$, such that no two adjacent vertices get same color.
- **3-Coloring** is a decision problem, where we want to determine whether a given graph G, can be colored using **at most 3 colors**.
- The above problem is *NP-Complete* because it belongs to NP and $3-SAT \leq_p 3-Coloring$.

1.3 Max Cut

- Given a graph $G = (V, E)$, a **cut** $C = (A, B)$ partitions the vertices of the graph into two disjoint subsets A and B. The cut is defined by a **cut-set**, which is a set of edges $\{(u, v) \in E \mid u \in A,\ v \in B\}$. The number of edges in the cut-set is also known as the **cardinality** of the cut.
- The aim of **Max Cut** problem is to decide whether there is a cut of size at least k in G, for an integer $k > 0$.
- The above problem is *NP-Complete* because it belongs to NP and $NAE3SAT\ \ (Not\text{-}all\text{-}equal\ \text{-}satisfiability) \leq_p Max\ Cut$.

1.4 Set Cover

- The input consists of a finite set S, and a collection C of subsets of S. A sub-collection $C' \subseteq C$ is termed as **set cover** of S, if the union of sets in C' equals S. In other words, every element of S, belongs to at least one member of the sub-collection C'.
- The problem at hand is to decide whether there exists a set cover C' for S such that $|C'| \leq k$, for some positive integer k.
- The above problem is *NP-Complete* because it belongs to NP and $Vertex\ Cover\ \ \leq_p Set\ Cover$.

1.5 Subset Sum

- We are given n integers $A = \{w_1, w_2, ..., w_n\}$ as input along with a **target sum** W (an integer).
- The **Subset Sum** problem is a decision problem where the objective is to answer whether any subset $B \subseteq A$ exists, such that $\sum\limits_{w_i \in B} w_i = W$.
- The above problem is *NP-Complete* because it belongs to NP and $SAT \leq_p Subset\ Sum$.

1.6 Max 3-D Matching

- Let us consider three finite and mutually disjoint sets: X, Y, Z. Now, we define $T \subseteq \{X \times Y \times Z\}$. Basically, every element of T is a triple (x, y, z) such that $x \in X, y \in Y, z \in Z$.
- A subset $M \subseteq T$ is known as **3-D Matching** if for any two distinct elements $u = \{u_X, u_Y, u_Z\} \in M$ and $v = \{v_X, v_Y, v_Z\} \in M$, the following condition is satisfied: $u_X \neq v_X$, $u_Y \neq v_Y$, $u_Z \neq v_Z$.
- Given a positive integer k, the objective of this problem is to decide whether there exists $M \subseteq T$ such that $|M| \geq k$.
- The above problem is *NP-Complete* because it belongs to NP and $3-SAT \leq_p Max\ 3-D\ Matching$.

1.7 Min Hitting Set

- The input consists of a finite set S, and a collection C of subsets of S. A **hitting set** for C is defined to be a subset $H \subseteq S$, such that H contains at least one element from each subset in C.
- Given a positive integer k, our goal is to determine whether there exists a hitting set $H \subseteq S$, such that $|H| \leq k$.
- The above problem is *NP-Complete* because it belongs to NP and $Vertex\ Cover\ \leq_p Min\ Hitting\ Set$.

1.8 Hamiltonian Cycle

- A path P in a graph G, is termed as a **Hamiltonian Path**, if it visits every vertex exactly once. A **Hamiltonian cycle**, also known as Hamiltonian circuit, is a Hamiltonian Path such that the last vertex of P is connected to the first vertex of P via an edge.
- The **Hamiltonian cycle** problem seeks to answer whether there exists a Hamiltonian cycle in G or not.
- The above problem is *NP-Complete* because it belongs to NP and $Vertex\ Cover\ \leq_p Hamiltonian\ Cycle$.

1.9 Integer Linear Programming

- It is a special case of the general **linear programming problem**, where some or all of the decision variables are restricted to be integers.
- It has various variants:
  (i) **Pure ILP**: Here, every variable is restricted to be an integer.
  (ii) **Mixed ILP**: While some variables are restricted to be integers, the others are permitted to take any value.
  (iii) **Binary ILP**: All variables are restricted to be only 0 or 1.
- The above problem is *NP-Complete* because it belongs to NP and $SAT \leq_p Integer\ Linear\ Programming$.

2 Problem 2

A problem X is said to be **NP Complete** if it satisfies the following conditions:

(i) X belongs to the class NP, i.e. $X \in NP$

(ii) A problem Y known to be NP-Complete, can be reduced to X in polynomial time, i.e. $Y \leq_p X$.

Let X denote our **Path Cover** problem. We are given an undirected graph G and an integer k, along with a set of m shortest paths $\mathcal{P} = \{P_1, P_2, ..., P_m\}$. The objective is to determine whether there exists at most k paths in $\mathcal{P}$ such that their union covers all the edges in G.

Part 1: $X \in NP$

Candidate Solution: For a graph $G = (V, E)$, we are given a set of paths $\mathcal{P}'$.

Claim 1: Any candidate solution for X has length of polynomial order in the input size.

$\mathcal{P}'$ is a set of at most k paths. Again, a path can have no more than $|E|$ edges. Hence, the total space required to represent a candidate solution is $O(kE)$. Here, k is bounded by m which itself can never exceed the maximum number of shortest paths possible in G. Thus, $m = O(\ \binom{V}{2}\ ) = O(V^2)$. As a result, the total space complexity of candidate solution $= O(V^2E)$, which is of polynomial order. Hence, Claim 1 holds.

Claim 2: Any candidate solution for X can be verified in polynomial time.

To verify a given possible solution, we need to iterate over all the paths in $\mathcal{P}'$, and while going over each path, mark the edges which get covered in the process. In the worst case, we might need to check each of the k paths in $\mathcal{P}'$. Again, each one of these paths can have at most $|E|$ edges. Therefore, the time to iterate over all the paths, is $O(kE) = O(V^2E)$. So, we can verify any solution in polynomial time. Hence, Claim 2 is also true.

From Clam 1 and Claim 2, we conclude that $X \in NP$.
