COMP 3804 — Solutions Assignment 4

Question 1: Write your name and student number.

Solution: Hinata Miyazawa, 20

Question 2: The *set cover problem* is defined as follows:

$$\textsc{SetCover} = \{(S, n, A_1, A_2, \ldots, A_m, K) : \begin{array}{l} S \text{ is a set of size } n, \text{ each } A_i \text{ is a subset of } S, \\ \exists I \subseteq \{1, 2, \ldots, m\} \text{ such that} \\ |I| = K \text{ and } \cup_{i \in I} A_i = S \ \}. \end{array}$$

Prove that SetCover is in **NP**.

Solution: The *verification algorithm* $\mathcal{V}$ does the following:

- It takes as input
  - a tuple $(S, n, A_1, A_2, \ldots, A_m, K)$ representing an input for SetCover,
  - a set I, representing the certificate.
- The verification algorithm does the following:
  - Check that $I \subseteq \{1, 2, \ldots, m\}$.
  - Check that $|I| = K$.
  - Check that $\cup_{i \in I} A_i = S$.
  - If all of these are correct, then it returns YES. Otherwise, it returns NO.

$$\begin{aligned} (S, n, A_1, \ldots, A_m, K) \in \textsc{SetCover} \quad &\Leftrightarrow \quad \text{there exists } I \text{ such that} \\ &\qquad I \subseteq \{1, \ldots, m\},\ |I| = K,\ \cup_{i \in I} A_i = S \\ &\Leftrightarrow \quad \text{there exists a certificate } I \text{ such that} \\ &\qquad \mathcal{V}(S, n, A_1, \ldots, A_m, K, I) \text{ returns YES.} \end{aligned}$$

The length of the certificate I is equal to K, which is at most m, which is at most the length of $(S, n, A_1, \ldots, A_m, K)$.

What is the running time of the verification algorithm:

- Checking that $I \subseteq \{1, 2, \ldots, m\}$ can be done in $O(|I| \cdot m) = O(m^2)$ time. (Of course, there are faster ways to do this.)
- Using a sorting algorithm, checking that $|I| = K$ can be done in $O(|I| \log |I|) = O(m^2)$ time.

- Checking that $\cup_{i \in I} A_i = S$ can be done in time proportional to

$$\sum_{i \in I} |A_i| \cdot |S|.$$

(Of course, there are faster ways to do this.) This is polynomial in the length of $(S, n, A_1, \ldots, A_m, K)$.

Question 3: *Los Tabernacos* is a famous poutine restaurant in Playa del Carmen, Mexico. The owners want to advertize their restaurant to all people ("users") on Instagram. For a given integer K, they ask K users to post a picture of the restaurant[1] on their account.

All users follow the Instagram etiquette: If user u posts a picture, then all users who follow u post a copy of this picture.

Can the owners of Los Tabernacos choose K users such that all Instagram users post a picture of the restaurant?

- Formulate this problem as a decision problem LosTabernacos on a graph.
- Prove that LosTabernacos $\leq_P$ SetCover, i.e., in polynomial time, LosTabernacos can be reduced to SetCover.

Solution: We define a directed graph $G = (V, E)$, where V is the set of all Instagram users. There is an edge (u, v) from u to v, if v follows u. For each vertex u, let $R(u)$ be the set of all vertices v such that there is a directed path from u to v. Note that $u \in R(u)$. Based on this, we get

$$\begin{aligned} \textsc{LosTabernacos} = \{(G, K) : \quad & G = (V, E) \text{ is a directed graph,} \\ & \exists I \subseteq V \text{ such that } |I| = K \text{ and } \cup_{u \in I} R(u) = V \ \}. \end{aligned}$$

Next we prove that LosTabernacos $\leq_P$ SetCover. Consider an input (G, K) for LosTabernacos. We map (G, K) to an input

$$f(G, K) = (S, n, A_1, A_2, \ldots, A_m, K)$$

for SetCover:

- Define $S = V$.
- Define $n = |V|$.
- Number the vertices of V as $u_1, u_2, \ldots, u_n$.
- Define $m = n$.
- For each $i = 1, 2, \ldots, n$, define $A_i = R(u_i)$.

[1] and offer them free poutine

- The value of K remains the same.

By construction, $(G, K) \in$ LosTabernacos is equivalent to $(S, n, A_1, A_2, \ldots, A_m, K) \in$ SetCover.

How much time is needed to compute $f(G, K)$: To compute $R(u_i)$, we do the following:

- For each vertex v, set $visited(v) = false$.
- Run algorithm Explore(u_i).
- Set $R(u_i)$ to the set of all vertices v for which $visited(v) = true$.

This takes $O(|V|+|E|)$ time per vertex. In total, this takes $O(|V|^2+|V|\cdot|E|) = O((|V|+|E|)^2)$ time, which is polynomial in the length of G.

Question 4: Let $G = (V, E)$ be an undirected graph. A *Hamilton cycle* is a cycle in G that contains every vertex exactly once. A *Hamilton st-path* is a path in G between the vertices s and t that contains every vertex exactly once.

Consider the problems

$$\textsc{HamiltonCycle} = \{G : \text{ graph } G \text{ contains a Hamilton cycle}\}$$

and

$$\textsc{HamiltonPath} = \{(G, s, t) : \text{ graph } G \text{ contains an } st\text{-Hamilton path}\}.$$

- Prove that HamiltonCycle $\leq_P$ HamiltonPath, i.e., in polynomial time, HamiltonCycle can be reduced to HamiltonPath.
- Prove that HamiltonPath $\leq_P$ HamiltonCycle, i.e., in polynomial time, HamiltonPath can be reduced to HamiltonCycle.

Solution: Note that the question does not specify if the vertices s and t can be equal. I will give two solutions, one where s and t can be equal, and one where s and t must be distinct.

We start with HamiltonCycle $\leq_P$ HamiltonPath. Consider an input $G = (V, E)$ for HamiltonCycle. We will map G to an input

$$f(G) = (G', s, t)$$

for HamiltonPath, such that G has a Hamilton cycle if and only if G' has an st-Hamilton path. From the construction, it will be clear that $f(G)$ can be computed in time that is polynomial in the length of G.

- If s and t can be equal: Define $G' = G$. Take an arbitrary vertex v in V. Define $s = t = v$. Note that a Hamilton cycle in G is exactly the same as an st-Hamilton path in G'.

- If s and t cannot be equal:
  - Take an arbitrary vertex v in V.
  - Introduce three new vertices v', s, and t.
  - Define $V' = V \cup \{v', s, t\}$.
  - The edge set E' of G' contains all edges of E. Additionally, E' contains the two edges $\{s, v\}$ and $\{t, v'\}$. Finally, for each edge $\{v, w\}$ in E, E' contains the edge $\{v', w\}$.

  Assume that G contains a Hamilton cycle. Traverse this cycle, starting at v, and let w be the last vertex on this cycle before returning to v. In G', this gives an st-Hamilton path: Start at s, go to v, follow the cycle until w, go to v', then go to t.

  Conversely, assume that G' contains an st-Hamilton path. The first edge on this path must be $\{s, v\}$, and the last edge must be $\{v', t\}$. Let $w \neq t$ be the vertex such that $\{w, v'\}$ is the edge on the path that takes us to v'. We get a Hamilton cycle in G: Start at v, follow the portion of the st-path from v until w, then go back to v.

Next we show that HamiltonPath $\leq_P$ HamiltonCycle. Consider an input (G, s, t) for HamiltonPath, where $G = (V, E)$. We will map (G, s, t) to an input

$$f(G, s, t) = G'$$

for HamiltonCycle, such that G has an st-Hamilton path if and only if G' has a Hamilton cycle. From the construction, it will be clear that $f(G, s, t,)$ can be computed in time that is polynomial in the length of (G, s, t).

- If $s = t$: We define $G' = G$. Note that an st-Hamilton path in G is exactly the same as a Hamilton cycle in G'.
- If $s \neq t$:
  - Introduce a new vertex x.
  - Define $V' = V \cup \{x\}$.
  - The edge set E' of G' contains all edges of E. Additionally, E' contains the two edges $\{s, x\}$ and $\{t, x\}$.

  An st-Hamilton path in G leads to a Hamilton cycle in G', by adding the edges $\{s, x\}$ and $\{x, t\}$ to this path.

  Conversely, consider a Hamilton cycle in G'. This cycle must contain the two edges $\{s, x\}$ and $\{x, t\}$. If we remove them, we get an st-Hamilton path in G.

Question 5: In the *longest path problem*, we are given an undirected graph $G = (V, E)$ in which each edge has a positive weight, two vertices s and t, and a number L. The question is whether or not G contains an st-path (i.e., a path between s and t) of length at least L. In such a path, any vertex cannot be visited more than once.

$$\textsc{LongestPath} = \{(G, s, t, L) : \text{ graph } G \text{ contains an } st\text{-path of length at least } L\}.$$

Prove that $\textsc{HamiltonCycle} \leq_P \textsc{LongestPath}$, i.e., in polynomial time, $\textsc{HamiltonCycle}$ can be reduced to $\textsc{LongestPath}$.

Solution: In the previous question, we have shown that $\textsc{HamiltonCycle} \leq_P \textsc{HamiltonPath}$. Since the relation $\leq_P$ is transitive, it is sufficient to show that $\textsc{HamiltonPath} \leq_P \textsc{LongestPath}$.

Consider an input (G, s, t) to $\textsc{HamiltonPath}$, where $G = (V, E)$. We have to map this to an input (G', s', t', L') to $\textsc{LongestPath}$ such that G contains an st-Hamiltonian path if and only if G' contains an st-path of length at least L'.

Note that the edges in G do not have weights, whereas the edges in G' do have weights. Here is the mapping:

- $G' = (V, E)$, i.e., G' has the same vertex and edge sets as G.
- Each edge of G' gets a weight of one.
- $s' = s$ and $t' = t$.
- $L' = n - 1$, where n is the number of vertices.

It is clear that the mapping can be computed in time that is polynomial in the size of (G, s, t).

First assume that G contains an st-Hamilton path. Since the number of edges on this path is $n - 1$, and in G' each edge has a weight of one, the graph G' contains an st-path of length L', which is at least L'.

Now assume that G' contains an st-path of length at least $L' = n - 1$. Since a path cannot contain more than $n - 1$ edges, the length of the st-path in G' has length exactly $n - 1$. Since all edges in G' have a weight of one, this st-path has exactly $n - 1$ edges and n vertices. Therefore, in G, this st-path is an st-Hamilton path.

Question 6: A Boolean formula φ, in the variables $x_1, x_2, \ldots, x_n$, is in *three conjunctive normal form* (3CNF), if it is of the form

$$\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_m,$$

where each *clause* C_i, $1 \leq i \leq m$, is of the form

$$C_i = l_1^i \vee l_2^i \vee l_3^i.$$

Each l_j^i is a *literal*, which is either a variable or the negation of a variable.

The *three-satisfiability problem* is defined as follows:

$$\text{3SAT} = \{\varphi : \varphi \text{ is in 3CNF-form and is satisfiable}\}.$$

A *vertex cover* of an undirected graph $G = (V, E)$ is a subset X of V such that for each edge $\{u, v\}$ in E, at least one of u and v is in X.

The *vertex cover problem* is defined as follows:

$$\text{VertexCover} = \{(G, K) : \text{ graph } G \text{ contains a vertex cover of size } K\}.$$

Prove that 3SAT $\leq_P$ VertexCover, i.e., in polynomial time, 3SAT can be reduced to VertexCover.

Solution: When I made this assignment, I did not realize that this follows from three results that were shown in class:

- We have shown that 3SAT $\leq_P$ IndependentSet.
- We have shown that IndependentSet $\leq_P$ VertexCover. This was based on the fact that X is a vertex cover of size K in the graph $G = (V, E)$ if and only if $V \setminus X$ is an independent set of size $|V| - K$ in the same graph.
- We have shown that the relation $\leq_P$ is transitive.