Assignment 4 - Problem 6.2, 6.7

Kousshik Raj (17CS30022)

08-10-2020

1 Problem 6.2

Using dynamic programming over subsets, show that the HAMILTONIAN CYCLE problem on n-vertex graph can be solved in time $2^n n^{O(1)}$

1.1 Solution

Let the given graph be $G = (V, E)$, where $|V| = n$. Now, we want to find whether a Hamiltonian Cycle exists in the graph or not. Choose an arbitrary vertex $v_0 \in V(G)$. Now, for a Hamiltonian Cycle to exist, there should exist a simple path of vertices $v_0, v_1, ..., v_{n-1}$ and $(v_0, v_{n-1}) \in E(G)$. We will use this fact to solve the problem using dynamic programming over subsets.

Fix v_0 as the starting point. Define $X[S, u]$, where $S \subseteq V(G), u \in S$ to be *true* when there is a simple path starting at v_0 and ending at u which includes all the vertices in S and has exactly $|S|$ vertices, otherwise $X[S, u]$ will be *false*. The recurrence relation will be

$$X[S,u] = \begin{cases} \textit{false} & \text{if } u \notin S \text{ or } v_0 \notin S \\ \textit{true} & \text{if } S = \{v_0\} \\ \vee\{X[S - \{u\}, v] : (u, v) \in E(G)\} & \text{otherwise} \end{cases}$$

The first two conditions are trivial, as there can be no path without the starting and ending points present, and you can always start and end at v_0. The last relation arises from the fact that if there exists a path starting at v_0, going through all the vertices in $S` = S - \{u\}$ and ending at v, and if $(v, u) \in E(G)$, then we can move along this edge to u, and thus we have a path that starts at v_0, goes through all the vertices in S and ends at u.

As we have previously seen, there exists a Hamiltonian Cycle in G iff there exists $v \in V(G)$ such that $X[V(G), v] = \textit{true}$ and $(v, v_0) \in E(G)$. Since there are $2^n \cdot n$ states, and each state can be computed in $O(n)$ time, the algorithm runs in $O(2^n n^2)$ time.

2 Problem 6.7

Given a directed graph G, a set of terminals $K \subseteq V(G)$ and a root $r \in V(G)$, DIRECTED STEINER TREE asks for a directed tree rooted at r such that every terminal in K is reachable from r on the tree. Obtain a $3^{|K|}n^{O(1)}$-time algorithm for DIRECTED STEINER TREE.

2.1 Solution

Given that G is a directed graph, let's define $D(u, v)$, where $u, v \in V(G)$, as the minimum cost of the path from u to v. If such a path doesn't exist, we can take $D(u, v) = \infty$. We can calculate $D(u, v)$ for all possible values in polynomial time using Floyd-Warshall algorithm. Now, we will look at a dynamic programming algorithm to calculate the minimum-weight directed Steiner Tree.

First, we will process the input graph G such that all terminal vertices are sinks. Define (G', K') from (G, K) such that, $V' = V \cup \{v'\}, E' = E \cup \{(v, v')\}, K' = \cup\{v'\}, \forall v \in K$, where v' is a new terminal vertex corresponding to the terminal v. If we assign the weight of the new edges as 0, it is obvious that the weights of both the minimum Steiner Trees are the same and moreover, there is a one to one correspondence between the minimum Steiner Trees of the instances (G, K) and (G', K'). Thus, we can now work under the assumption that there is no outgoing edge from the terminal vertices in the instance (G, K).

Let $X[S, v]$, where $S \subseteq K, v \in V(G) - K$, denote the weight of the minimum Steiner Tree such that all the vertices(terminal) in S are reachable from v. We finally need the value $X[K, r]$ from the table. It is to be noted that, if the value $X[K, r]$ is ∞, then a Steiner Tree doesn't exist for the given instance. The recurrence relation is

$$X[S, v] = \begin{cases} 0 & \text{if } S = \phi \\ D(v, t) & \text{if } |S| = 1, t \in S \\ \min\limits_{\forall S' \subseteq S, u \in V - K} \{X[S', u] + X[S - S', u] + D(v, u)\} & \text{otherwise} \end{cases}$$

The first two cases are trivial. We will now prove the third scenario, where $|S| > 1$. Let, H be a graph where all the terminals in S are reachable from v, H_1 be the minimum Steiner Tree where all the terminals in S' are reachable from u, H_2 be the minimum Steiner Tree where the terminals in $S - S'$ are reachable from u, and P be a path from vertex v to u. It is obvious that, the graph resulting from $H_1 \cup H_2 \cup P$ can be taken as H, as we can reach all the terminals in S from u and we can reach u from v, and thus by extension we can reach all terminals in S from v. Therefore, $w(H) \leq w(H_1) + W(H_2) + w(P)$, which means, $X[S, v] \leq \min\limits_{\forall S' \subseteq S, u \in V - K} \{X[S', u] + X[S - S', u] + D(v, u)\}$.

For, the other direction, consider the minimum Steiner Tree H, where all terminals in S are reachable from v. Consider a vertex u reachable from v and has at least 2 outgoing edges, and among them, consider the closest one to v.

Such a vertex exists because, $|S| > 1$ and no outgoing edges are possible from the terminals. Since there are at least 2 outgoing edges from u, consider H_1 as a graph rooted at one of its children along with u and, H_2 as the graph rooted at u excluding the edges of H_1. Let $S' = S \cap H_1$, then $S \cap H_2 = S - S'$, as there are no terminals in the path from v to u. It is obvious that both S' and $S - S'$ are non-empty, as otherwise, it contradicts the assumption that H is the minimum Steiner Tree. Furthermore, since H is optimal, we can say that $X[S, v] \geq \min_{\forall S' \subseteq S, u \in V-K} \{X[S', u] + X[S - S', u] + D(v, u)\}$, which arises from the parts H_1, H_2 and the path from v to u.

Thus, from the above two inequalities proven, we can say that, whenever $|S| > 1, X[S, v] = \min_{\forall S' \subseteq S, u \in V-K} \{X[S', u] + X[S - S', u] + D(v, u)\}$. There are $2^{|K|} \cdot n$ states, and for each state, we iterate through all of the subsets of the terminals we are considering. Therefore, by the binomial theorem, we have the running time of our algorithm as $O(3^{|K|} n^{O(1)})$.