Algorithm HW14

1. Show that any comparison-based algorithm needs (n log n) time to solve the following problem: Given n points in the 2D plane, construct a tree of minimum total length whose vertices are the given points and the edge length of an edge is the Euclidean distance between the two end points of the edge.

2. **Exercises 34.1-5**
   Show that if an algorithm makes at most a constant number of calls to polynomial-time subroutines and performs an additional amount of work that also takes polynomial time, then it runs in polynomial time. Also show that a polynomial number of calls to polynomial-time subroutines may result in an exponential-time algorithm.

3. **Exercises 34.1-6**
   Show that the class **P**, viewed as a set of languages, is closed under union, intersection, concatenation, complement, and Kleene star. That is, if $L_1, L_2 \in$ **P**, then $L_1 \cup L_2 \in$ **P**, $L_1 \cap L_2 \in$ **P**, $L_1L_2 \in$ **P**, $\overline{L_1} \in$ P, and $L_1{}^* \in$ **P**.

4. **Exercises 34.2-3**
   Show that if HAM-CYCLE ∈ P, then the problem of listing the vertices of a hamiltonian cycle, in order, is polynomial-time solvable.
   (Note 1: HAM-CYCLE is defined as "Does a graph G have a Hamiltonian cycle?")
   (Note 2: "HAM-CYCLE ∈ P" means that HAM-CYCLE is polynomial-time solvable.)

5. **Exercises 34.2-7**
   Show that the Hamiltonian-path problem can be solved in polynomial time on directed acyclic graphs. Give an efficient algorithm for the problem.

6. **Exercise 34.2-1**
   Consider the language GRAPH-ISOMORPHISM = {<G_1, G_2> : G_1 and G_2 are isomorphic graph}. Prove that GRAPH-ISOMORPHISM ∈ NP by describing a polynomial-time algorithm to verify the language.

7. **Exercise 34.1-2**
   Consider the problem of finding the longest simple cycle in an undirected graph. Give a related decision problem.Show the decision problem is an NP problem.