Problem 8.1. (30 points) In the following, let G be an undirected graph, let s and t be two nodes of G, and let k be a natural number. A path of G is *simple* if no node occurs more than once along the path.

$$\begin{aligned}\textsc{Spath} &= \{\langle G, s, t, k\rangle \mid G \text{ contains a simple path of length at } \textit{most } k \text{ from } s \text{ to } t\}\\ \textsc{Lpath} &= \{\langle G, s, t, k\rangle \mid G \text{ contains a simple path of length at } \textit{least } k \text{ from } s \text{ to } t\}\end{aligned}$$

One of the two problems is in P, the other one is NP-complete. Which is which?

- For the problem in P, give a polynomial-time algorithm.
- For the NP-complete problem, give a certificate that can be verified in polynomial time and a reduction from a known NP-hard problem.

Hint. For a known NP-hard problem, use either CLIQUE or the undirected graph version of HAMPATH. That is, make use of the fact that the following two problems are NP-hard:

$$\begin{aligned}\textsc{UHamPath} &= \{\langle G, s, t\rangle \mid \text{the undirected graph } G \text{ has a Hamiltonian path from vertex } s \text{ to vertex } t\}.\\ \textsc{Clique} &= \{\langle G, k\rangle \mid \text{the undirected graph } G \text{ has a clique with } k \text{ vertices}\}.\end{aligned}$$

Solution: Spath is in P, and Lpath is NP-complete.

1. Spath $\in$ P.

   Consider the following **deterministic** algorithm.

   Input: $\langle G, s, t, k\rangle$
   1. Mark the node s.
   2. for $i = 1$ to k do
      for all edge (a, b) in G do
        if a is marked then mark b
   3. If t is marked then ACCEPT else REJECT

   Let $n = |\langle G, s, t, k\rangle|$. Step 1. and 3. are both $O(n)$. The loop in 2. is $k \cdot O(n)$. Therefore the overall complexity of the algorithm is $O(k \cdot n)$ and Spath $\in$ P.

2. Lpath is NP-complete.

   - We first prove that Lpath $\in$ NP.

     We use the path as the certificate. The following **non-deterministic** Turing machine **decides** Lpath.

     Input: $\langle G, s, t, k\rangle$
     1. Nondeterministically select a path c in G of length at least k.
     2. Test whether
        c start with s and ends with t.
        c does not visit the same node twice.
     3. If all tests pass then ACCEPT else REJECT

Let $n = |\langle\langle G, s, t, k\rangle\rangle|$. It is easy to see that all the tests can be performed in $O(n)$. Thus the above decider is **non-deterministic polynomial time** Turing decider, and therefore $\text{LPATH} \in \text{NP}$.

Alternatively, the following machine is a **deterministic verifier** for LPATH. We use the path as the certificate.

Input: $\langle\langle G, s, t, k\rangle, c\rangle$
1. Test whether
   - c is a path in G of length at least k.
   - c start with s and ends with t.
   - c does not visit the same node twice.
2. If all tests pass then ACCEPT else REJECT

Let $m = |\langle\langle G, s, t, k\rangle, c\rangle|$. It is easy to see that all the tests can be performed in $O(m)$ and therefore $\text{LPATH} \in \text{NP}$.

- We next show that LPATH is NP-hard by proving that $\text{UHAMPATH} \leq_p \text{LPATH}$. We give a function f that maps every triple $\langle G, s, t\rangle$ to a quadruple $\langle G, s, t, n-1\rangle$, where n is the number of nodes in G. We need to show that:

  (a) for every triple $\langle G, s, t\rangle$, $\langle G, s, t\rangle \in \text{UHAMPATH}$ iff $\langle G, s, t, n-1\rangle \in \text{LPATH}$,

  (b) f is a polynomial time computable function.

  Counting the number of nodes of a graph clearly takes an amount of time that is polynomial with the size of the graph, so f is polynomial, and we are done with the second item. Let us now prove the first of the items above:

  - ($\Rightarrow$) If $\langle G, s, t\rangle \in \text{UHAMPATH}$ then, by definition of Hamiltonian Path, there is in G a simple path from s to t that touches all the nodes in G. Of course, the length of this path is $n-1$. Therefore, G contains a simple path of length at least $n-1$ from s to t. This means that $\langle G, s, t, n-1\rangle \in \text{LPATH}$.
  - ($\Leftarrow$) If $\langle G, s, t, n-1\rangle \in \text{LPATH}$ then, by definition, G contains a simple path of length at least $n-1$ from s to t. Since G has n nodes, a path of length $n-1$ must visit all the nodes of the graph. Moreover, since the path is simple, each node is visited at most once. This means that the path from s to t is a Hamiltonian path, and therefore $\langle G, s, t\rangle \in \text{UHAMPATH}$.