

COSC341 - Assignment 3

Tom Berg, Elijah J. Passmore, Dan Skaf

May 23, 2021

Contents

1	Longest Path Problem	1
1.1	Simple Path	1
1.2	Longest Simple Path and Decision Version	2
1.3	NP	2
2	NP-Complete	2
2.1	Hamiltonian Path	3
2.2	Reduction to Longest Path Problem	3
2.3	Correctness	3
2.3.1	A Longest Path Exists	3
2.3.2	A Hamiltonian Path Exists	4
3	Conclusion	4

1 Longest Path Problem

The **Longest Path Problem** is for finding the maximum length simple path in a graph, $G(V, E)$ with $|V| = n : n \in \mathbb{N}$. The graph is undirected and all edges weights are assumed to be 1.

1.1 Simple Path

Suppose we want the **simple path** between two vertices s and t . A simple path is one which every node in the path is never visited more than once and that there is a path between two consecutive nodes.

Therefore, a simple path between two nodes $s, t \in V$ is the sequence of vertices $(v_1, v_2, v_3, \dots, v_k)$ that satisfies these conditions:

- $s = v_1$ and $t = v_k$
- Each consecutive nodes (v_i, v_{i+1}) there is an edge $e = (v_i, v_{i+1}) \in E$
- No node appears more than once in the sequence.

1.2 Longest Simple Path and Decision Version

The figure below shows the longest simple path in blue, beginning in the upper left:

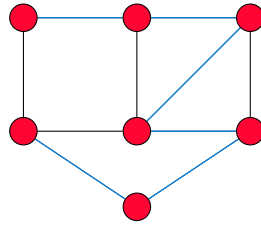


Figure 1: Longest simple path

In the decision version of the Longest Path Problem, we input a graph G , and an integer k , and output a **yes** or **no** if a simplest path of **at least** length k exists in the graph.

1.3 NP

A problem is in **NP** if we can verify the solution in polynomial time, and if a corresponding certificate can be produced from the decision version of the Longest Path Problem. If we can verify this certificate in polynomial time, then the Longest Path Problem is in **NP**. The certificate is a path of at least length k with no duplicate nodes.

To check that the path has no duplicate nodes we can iterate over each node in the path for every node, $O(k^2)$ complexity. To check the length of the path we just need to count each node and check it is greater than or equal to k , this operation has a $O(k)$ complexity.

In some implementations, getting the length of a list is constant complexity $O(1)$. In most cases and the worst case we can check a certificate with is $O(n^2)$ complexity, therefore the Longest Path Problem is in **NP**.

2 NP-Complete

If a problem, X , is in **NP-Complete** and X polynomially reduces to another problem Y in **NP**, then Y is also in **NP-Complete**.

We have shown that the Longest Path Problem is in NP, all that is left is to show that another problem in NP-Complete polynomial reduces to the Longest Path Problem.

2.1 Hamiltonian Path

A **Hamiltonian path** is a sequence that visits each node **exactly** once. The **Hamiltonian Path Problem** is a known NP-Complete problem that asks if a Hamiltonian path exists for a given graph.

2.2 Reduction to Longest Path Problem

With any instance of the Hamiltonian Path Problem of an undirected graph of size n , we can instead ask if a longest path of size $k = n - 1$ exists.

When we set k to $n - 1$, we are asserting that the longest path is equal to the Hamiltonian path.

2.3 Correctness

To prove our reduction we must show that:

Correct decision version of Hamiltonian path

\updownarrow

Correct decision version of Longest Path

That is, the instance of the Hamiltonian Path Problem exists only if a longest path of $k = n - 1$ exists. We will prove this in both directions from the definitions.

2.3.1 A Longest Path Exists

- If a longest path of size $k = n - 1$ exists, there is a simple path that passes each node exactly once.
- This simple path is a Hamiltonian path, and if a longest path does not exist there cannot be a Hamiltonian path.

2.3.2 A Hamiltonian Path Exists

- If a Hamiltonian path exists, there is a simple path of length $n - 1$.
- This is the maximum possible length of any simple path in the graph, therefore it is a longest path of size $k = n - 1$.
- If a Hamiltonian path does not exist, it is impossible for a longest path of $n - 1$ to exist, illustrated by a disconnected graph in *fig.2*

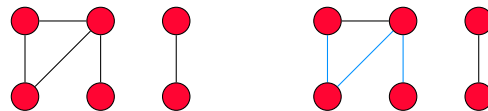


Figure 2: Graph without a Hamiltonian path, longest path on the right.

Therefore we have proved that our reduction is correct.

3 Conclusion

We have shown that the decision version of any Longest Path Problem has a certificate that is checked in polynomial time, therefore it is in NP. We showed a reduction from the Hamiltonian Path Problem to the Longest Path Problem, and proved the correctness of the reduction. Therefore, we have proved that the Longest Path Problem is also NP-Complete ■