Solution to the NP Completeness Practice Problem : A2 and B1

1 Problem Definition

Let G represent an *undirected* graph.

Prove that the problem of deciding whether G contains a simple path of length at least k from vertex a to vertex b is NP-complete.

Hamiltonian path in a graph goes through each vertex of the graph exactly once. We already know that the problem of deciding whether a graph contains a Hamiltonian path is NP-complete (NPC). You can use this assumption for proof.

To show that any problem A is NP-Complete, we need to show four things:

1. There is a non-deterministic polynomial-time algorithm that solves A, i.e., $A \in NP$.
2. Any NP-Complete problem B can be reduced to A.
3. The reduction of B to A works in polynomial time.
4. The original problem A has a solution if and only if B has a solution.

2 Preliminaries

For this kind of proof, we basically need to do two things:

- Prove $A \in NP$, in other words, prove that a solution (or *certificate*) can be verified in polynomial time.
- Prove $B \leq_P A$, where B is known to be a NP-Complete problem.

Point 1 of the problem definition is another way of saying a solution can be verified in polynomial time (see the first two paras here : `https://en.wikipedia.org/wiki/NP_(complexity)` if you are more curious).

The other three points are related to finding a reduction of a known NPC problem to your problem (which you are trying to prove is NPC) in polynomial time. You have to first describe a reduction process (Point 2) and then explain problem A has a solution *if and only if* problem B has a solution (Point 4). Your process also must run in polynomial time (Point 3).

Note that in general, for this step you can use any known NPC problem and also there can be more than one such polynomial time reduction algorithm for a particular known NPC problem. So other proofs not following my procedures would be perfectly fine as long as they follow the above steps, I am just showing one possible proof.

3 The Proof

For the actual proof, we go through the two steps as mentioned in the previous section.

3.1 Verification in Polynomial time

If we have a simple path (i.e. no repeating vertices, thus no cycles) from a to b, we may verify it by first checking if all the edges in the path belongs to the graph and then checking if the no. of edges in the path is $\geq k$. This checking can be done in polynomial time.

3.2 Reduction of a known NPC algorithm

Now we have to prove that $B \leq_P A$, where B is known to be a NP-Complete problem.

I want to first prove that the problem of whether there is a Hamiltonian path from two vertices a and b in Graph G is a NPC problem. The proof is:

- Verification of whether a path from a to b is a Hamiltonian path: check whether the path contains all the vertices in G exactly once, this can be done in polynomial time.

- The known NPC problem of whether a graph contains a Hamiltonian path can be reduced to this problem: Check whether there is a Hamiltonian path from u to v (this is our problem) for all possible vertex pairs (u, v) in the graph. If there is one, we get a Hamiltonian path. If we don't find a Hamiltonian path for any of the vertex pairs, there is no Hamiltonian path in the graph. So the reduction is complete.

  Also as the number of possible vertex pairs in a graph with N nodes is $\frac{N(N-1)}{2}$, or of polynomial order, this reduction is also is in polynomial time.

Thus the problem of whether there is a Hamiltonian path from two vertices a and b in Graph G is a NPC problem.

Now for the actual problem in hand, if we can decide whether G contains a simple path of length at least k from vertex a to vertex b is NP-complete, we can also decide whether G contains a simple path of length $N-1$ from vertex a to vertex b, where N is the no. of nodes in the graph (if the latter is NPC, which is just a special case of the former, the former is also NPC. Also observe that there can be no simple path from a to b greater than of length $N-1$). But a simple path of length $N-1$ going from a to b in G must be a Hamiltonian path, as it must visit all N vertices and a simple path cannot visit the same vertex twice.

So, we decide whether G contains a simple path of length $N-1$ from vertex a to vertex b. If so, there is a Hamiltonian path from a to b in graph G. If not, there is no Hamiltonian path from vertex a to vertex b in G. So we have reduced a known NPC problem to ours.

The reduction is also in polynomial time (we just set $k = N-1$, where N is the number of nodes in graph G).

Thus our proof is complete.

4 Some Extra Comments

You can also prove the given problem is NPC by finding a reduction algorithm from the *Hamiltonian cycle* problem too, however the reduction process would be much more complex. If you want to get an intuition of how to do this, you can visit this link:

`https://en.wikipedia.org/wiki/Hamiltonian_path_problem`

See the "Reduction between the path problem and the cycle problem" section. The second point would give you an idea. You can try to find out the reduction algorithm for your recreation, but it is not needed hopefully :p.