

Question 5

We can show that any algorithm determining if a graph has a tonian path is in NP using the verifier-based definition of NP.

If we take the tonian path as the certificate for the problem. We can create a verifier that walks the tonian path, and at each vertex makes sure that it hasn't been seen before. Walking the path takes $O(n)$ time and checking that the size of the path is at least half the number of vertices is also $O(n)$. Thus, the verifier runs in polynomial time and this problem is in NP.

To show this algorithm is NP-complete, we must also show that it is NP-hard.

We can use HamiltonianPath problem to compute tonian path, but we must show that it's possible to do the reverse in order to show that the tonian path problem is also NP-hard.

Since there are n possible endpoints to the Hamiltonian path in a graph, we can iterate through all of them in $O(n)$ time. On each of the iterations, add a "star" like structure with $n + 1$ vertices connected to the possible end point. Running the tonian path algorithm for each one of these constructed graphs will return whether or not there is a tonian path that uses all of the original vertices plus one extra. This tonian path will be of $n + 1$ length, but one will be added, thus if there is a path, it must be of length n in the original graph, and be a Hamiltonian path. This proves the correctness of the reduction and, consequently, the NP-hardness of tonian path.

Since tonian path is both NP and NP-hard, it is NP-complete as well.