CSCI 570 - HW 10

1) It is in NP because we can get a certificate in polynomial length.

Certificate: get a path of m edges that add up to at least the value k.

Certifier: check that all the edges m add up to at least k.

It is not in P because since it is not specified it is acyclic, then it has a cycle. Therefore, the running time to find the solution is not polynomial.

2) False, decision problems from NP-hard would not be solved in polynomial because they do not belong to NP. Besides, the only thing that should be polynomial is the optimization problem.

3) True, A reduces to B. Therefore, you are able to interpret A as an instance of B. The new problem is a hard problem, NP, to solve.

4) Certificate: get all the values between 1 and the actual number (excluding 1 and the actual number) to use them as divisors.

Certifier: check if any of the divisors can divide the target number. If yes, then it is composite. If not, then it is a prime number.

Yes, it is an NP.

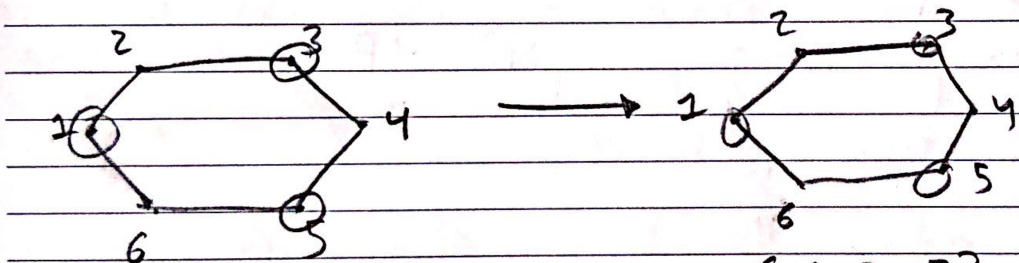
6) 1.- Prove that $X \in NP$.
2.- Choose a problem Y that is known to be NP-complete.
3.- Prove that $Y \leq_p X$.

Even degree vertices

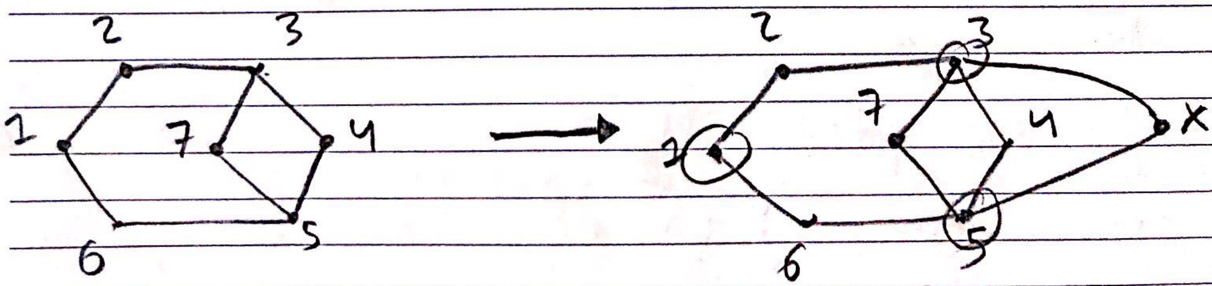
Certificate: get every edge $e \in E$ to have at least one end in S.

Certifier: check every edge to make sure at least one end is in S.

I know vertex cover is an NP-complete problem, so I will use it to prove $Y \leq_p X$.



$\{1, 3, 5\}$



$\{1, 3, 5\}$

Traverse every node and check if it is odd or even degree. If it is odd, then x can be added in polynomial time.

7) Loop through the graph G with the Hamiltonian cycle and delete the edges that do not belong to the cycle. Once all the non-cycle edges are removed, just list the remaining nodes in order of the remaining directed graph, O(e).

5) True. Proof: Reduce 3-SAT to an instance of Hamiltonian Cycle. Use the instance of the Hamiltonian Cycle and use the same procedure used in problem 7 to check if the cycle is still there after removing an edge. The final graph G' with a Hamiltonian Cycle can be reduced to a 3-SAT in polynomial time. Therefore, there is a polynomial time algorithm that does this procedure.