Ex 1

We will describe a decision problem fitting to the optimization problem seen in the given question.

Input:

* Graph .
* .
* A path length .

Output:

If there is a s.t and also .

Section 2

Now we will prove that the decision problem we have formulated, is in NP. In order to prove so, we need to find a fitting function that can be computed in a polynomial time by a non-deterministic Turing machine.   
First we find the shortest path between and using Dijkstra. Now we set .   
Next, our function takes as input a graph G and and traverse on the graph from to any adjacent node to , and from each of those to their adjacent nodes (giving that this node not already on the current path) and so on and so forth. Each path that reaches to and the current path is different from we add it to , and set  
 .   
Now we prove that our function run in a polynomial time. We can see that the length of the longest path between two vertices in the graph that our function would check is as long as . And is finite.

Now we will show, that given a witness, we can verify its correctness in a polynomial time.   
Given a witness We can check in a polynomial time if the path is legal () , .