Homework 9
Due November 27, 2019

The *Longest Cycle* problem (LONG-CYCLE) accepts a vertex in graph *G*, along with a length *k*, and returns whether the graph contains a cycle of length *k* or more that includes *v*.

The *Longest Path* problem (LONG-PATH) accepts two vertices *u* and *v* in a graph *G* and a distance *k*, and it returns whether *G* contains a path *k* of length or longer between *u* and *v*.

1. If $LONG - CYCLE \in NP - Hard$ and $LONG - PATH \in NP$, which of the reductions below would be used to prove the $LONG - PATH \in NP - Complete$? Assume that both reductions are correct.

Reduction One

```
Input: G = (V, E): graph with n vertices and m edges
Input: n, m: order and size of G
Input: v: vertex of G
Input: k: cycle length
Output: LONG-CYCLE(G, v, k): whether v is part of a cycle of length
        k or greater in G
Algorithm: ReductionOne
for u ∈ N(v) do
    G.RemoveEdge(u, v);
    if LONG-PATH(G, u, v, k − 1) then
        return true;
    end
    G.AddEdge(u, v);
end
return false;
```

Reduction Two

```
Input: G = (V, E): graph with n vertices and m edges
Input: n, m: order and size of G
Input: u, v: two vertices of G
Input: k: positive integer
Output: LONG-PATH(G, u, v, k): whether G contains a path of length
        k or longer between u and v
Algorithm: ReductionTwo
H = Graph(2n);
Name n vertices of H the same as G;
Name other n vertices x_1, x_2, ..., x_n;
for j ∈ G do
    for k ∈ N(i) do
        H.AddEdge(j, k);
    end
end
H.AddEdge(u, x_1);
for i = 1 to n − 1 do
    H.AddEdge(x_i, x_{i+1});
end
H.AddEdge(x_n, v);
return LONG-CYCLE(H, u, k + n + 1);
```

Reduction One

2. What is the worst-case time complexity of your chosen reduction? You may use LP to represent the time required to solve LONG-PATH and LC to represent the time required to solve LONG-CYCLE.

LC = O(m) · O(LP)

3. Use your answer to the previous question and the assumptions that $LONG - CYCLE \in NP - Hard$ to prove $LONG - PATH \in NP - Complete$.

To prove LP is NP-complete, we must prove LP ∈ NP and LP ∈ NP-hard. Because in problem 2, we reduced LC (NP-hard, given) to LP, we know LP is NP-hard. It was also given that LP ∈ NP, from problem 1. Therefore, since LP is NP, and NP-hard, LP is in NPC.