Question#3

Part 1:

Counter example detail:

Consider the input with size 5 vertex as:

(v0,v1),(v0,v2),(v1,v4),(v2,v3),(v3,v4)

The output in the book will be :(v0,v1,v4) which is 2

But the actual answer: (v0,v2,v3,v4) which is 3

Counter example argument:

The algorithm only choose the lowest number in the outgoing edges and only go once which will make this a false algorithm. There might be the cases like above which the second edges that w goes to is the smallest but it only connects to one point which is the destination. Therefore, this algorithm miss the cases where it goes to other point which is not the smallest, but the path might go through more vertex.

Part 2

Algorithm idea:

This is just a simple implementation of BFS algorithm. Put the initial vertex in a queue and run BFS, while the queue is not empty, every time check the path is greater than the path before+1, then set the path value to the larger one. It should be returning the right number of output.

Algorithm detail:

L = 0

AllLength = [0]\*len(|V|)

Queue.push(v0)

While queue is not empty:

Node = Queue.pop()

For item in outgoing\_edges[node]:

If alllength[item]<alllength[node]+1:

Alllength[item]= alllength[node]+1

Queue.push(item)

If alllength[item]>L:

L = alllength[item]

Return L

Proof idea:

Proof idea of BFS, since the graph is an ordered graph, we can be sure that there wont be any cycle in the graph and all nodes only connect forward. Therefore, the end vertex which is vn must has no outgoing\_edge.

Proof detail:

Every time of my iteration, I check the vertex’s outgoing\_edges to make sure the value that this node is going to attend doesn’t had a greater number than it pretended to have. So my algorithm will come to the end if the queue has nothing in there. Also I also keep updating the greatest number which is L in the end. The L will never pass |V| since the graph is ordered and the largest possible path of the graph is only be |V|-1. So my algorithm will output the right answer.

Runtime Algorithm:

BFS: O(n+m)

Source: the lecture note, BFS algorithm