1857 Largest color value in a directed graph

Initial thinking

Step1 : detect for cycles

do topological sort

if len(topo-sort) = len(colors)

no cycle

else cycle

return -1

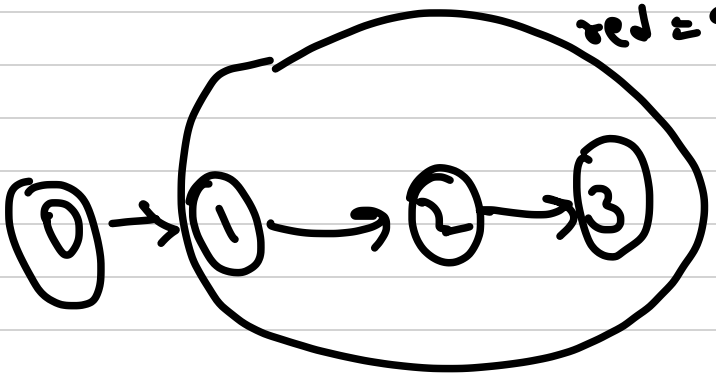
Step2 Thinking this as a dp problem

dp[node][clr] = largest color value of path starting from node, and having color = 'clr'

```
def solve_dp(node, color):

    ans = 0
    for nei in graph[node]:
        if colors[nei] >= color:
            ans = max(ans, 1 + solve_dp(nei, color))

        else:
            ans = max(ans, solve_dp(nei, color))

    return ans
```

step3: Find nodes with 0 indegree

Why?



Say for path 1-2-3 red = 2

then considering path 0-1-2-3 will have red ≥ 2

basically nodes with 0 indegree can be considered as start of paths(longest)

```
nts = [ ]  # nodes with 0 indegree

result = 0

for x in range(len(nts)):
    for clr in all_colors:
        result = max(result, solve_dp(nts[x], clr))

return result
```