207.LeetCode

**Course Schedule**

**Algorithm-Khans Algorithm**

1)First build the graph

2)We use restack to check the infinite loops

3)After building graph check the degree of the nodes

4)if any degree is zero insert in the queue and the start BFS using the all the inserted elements.

4)When ever we BFS happens we increase the counter to track the number of courses

5)At last we check the condition (it should be equal to numCourses

class Solution {

public boolean canFinish(int numCourses, int[][] prerequisites)

{

ArrayList<ArrayList<Integer>> graph=new ArrayList<>();

for(int i=0 ;i<numCourses ;i++)

{

graph.add(new ArrayList<>());

}

for(int[] edge :prerequisites)

{

graph.get(edge[0]).add(edge[1]);

}

int[] degree=new int[numCourses];

for(int i=0 ;i<numCourses ;i++)

{

ArrayList<Integer> temp=(ArrayList<Integer>)graph.get(i);

for(int x:temp)

{

degree[x]++;

}

}

int count=0;

Queue<Integer> q=new LinkedList();

for(int i=0 ;i<numCourses ;i++)

{

if(degree[i]==0)

{

q.add(i);

}

}

while(!q.isEmpty())

{

count++;

int tempo=q.poll();

for(int i :graph.get(tempo))

{

if(--degree[i]==0)

q.add(i);

}

}

if(count==numCourses)

return true;

return false;

}

}