**743. Network Delay Time**

**Algorithm -Bellaman Ford**

**(helps to detect cyclic infinite loops)**

**When they mention cyclic loops use bellaman ford or flyold marshall**

class Edge

{

int s,dfs,tfs;

Edge(int s,int dfs,int tfs)

{

this.s=s;

this.dfs=dfs;

this.tfs=tfs;

}

}

class Solution {

public int networkDelayTime(int[][] times, int N, int K)

{

if(N<=0 || times.length==0 || N<=0)

return -1;

ArrayList<Edge> graph=new ArrayList();

for(int i=0;i<times.length ;i++)

{

graph.add(new Edge(times[i][0],times[i][1],times[i][2]));

}

int[] dist=new int[N+1];

for(int i=1 ;i<dist.length ;i++)

{

dist[i]=Integer.MAX\_VALUE;

}

dist[K]=0;

for(int i=1; i<N ;i++)

{

for(Edge temp:graph)

{

int source=temp.s;

int dest=temp.dfs;

int time=temp.tfs;

if(dist[source]!=Integer.MAX\_VALUE && dist[source] + time < dist[dest] )

{

dist[dest]=dist[source] + time;

}

}

}

int max=Integer.MIN\_VALUE;

for(int i=0 ;i<dist.length ;i++)

{

if(dist[i]==Integer.MAX\_VALUE)

{

return -1;

}

max=Math.max(max,dist[i]);

}

return max;

}

}