Number of Provinces

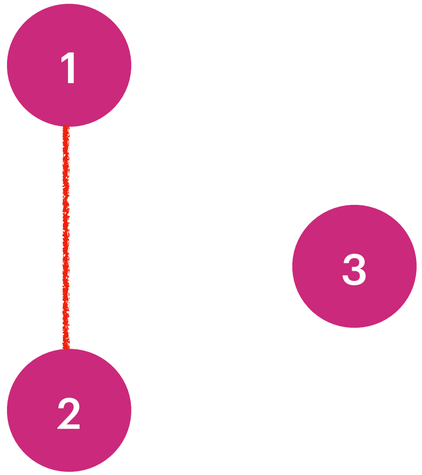
There are n cities. Some of them are connected, while some are not. If city a is connected directly city b, and city b is connected directly with city c, then city a is connected indirectly with city c.
A province is a group of directly or indirectly connected cities and no other cities outside of the group.
You are given an n x n matrix isConnected where isConnected[i][j] = 1 if the ith city and the jth city are directly connected, and isConnected[i][j] = 0 otherwise.
Return the total number of provinces.

Input: isConnected = [[1,1,0],[1,1,0],[0,0,1]]
Output: 2

Input: isConnected = [[1,0,0],[0,1,0],[0,0,1]]
Output: 3

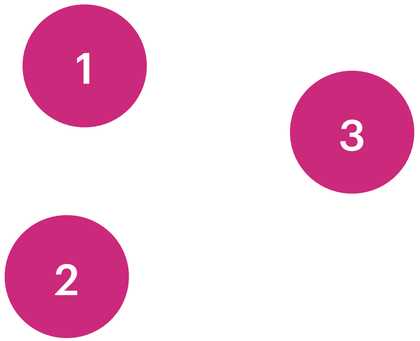
Input: isConnected = [[1,1,0],[1,1,0],[0,0,1]]
Output: 2

Count = 3
Count - - = 2



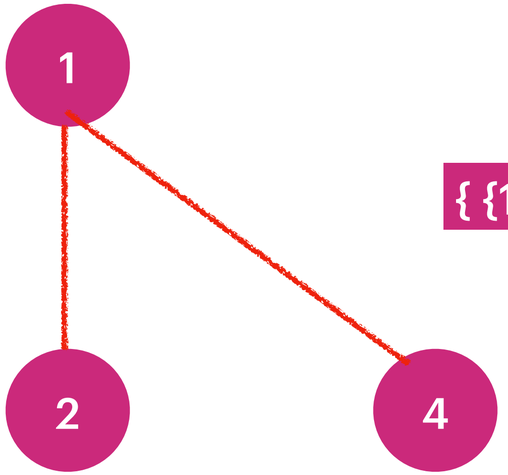
	C0	C1	C2
	[0,0]	[0,1]	[0,2]
R0	1	1	0
	[1,0]	[1,1]	[1,2]
R1	1	1	0
	[2,0]	[2,1]	[2,2]
R2	0	0	1

Input: isConnected = [[1,0,0],[0,1,0],[0,0,1]]
Output: 3



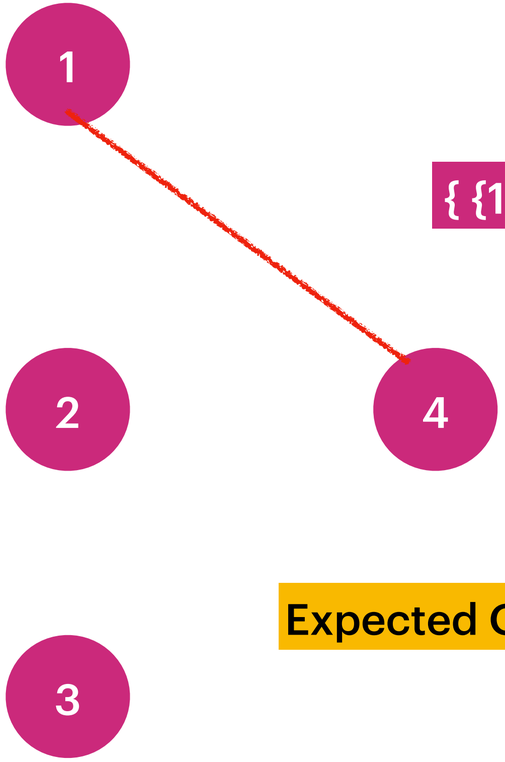
	C0	C1	C2
	[0,0]	[0,1]	[0,2]
R0	1	0	0
	[1,0]	[1,1]	[1,2]
R1	0	1	0
	[2,0]	[2,1]	[2,2]
R2	0	0	1

Count = 3
Count - - = 2



{ {1,1,0,1 }, {1,1,0,0},{0,0,1,0},{1,0,0,1} }

Expected Output : 2



{ {1,0,0,1 }, {0,1,0,0},{0,0,1,0},{1,0,0,1} }

Expected Output : 3

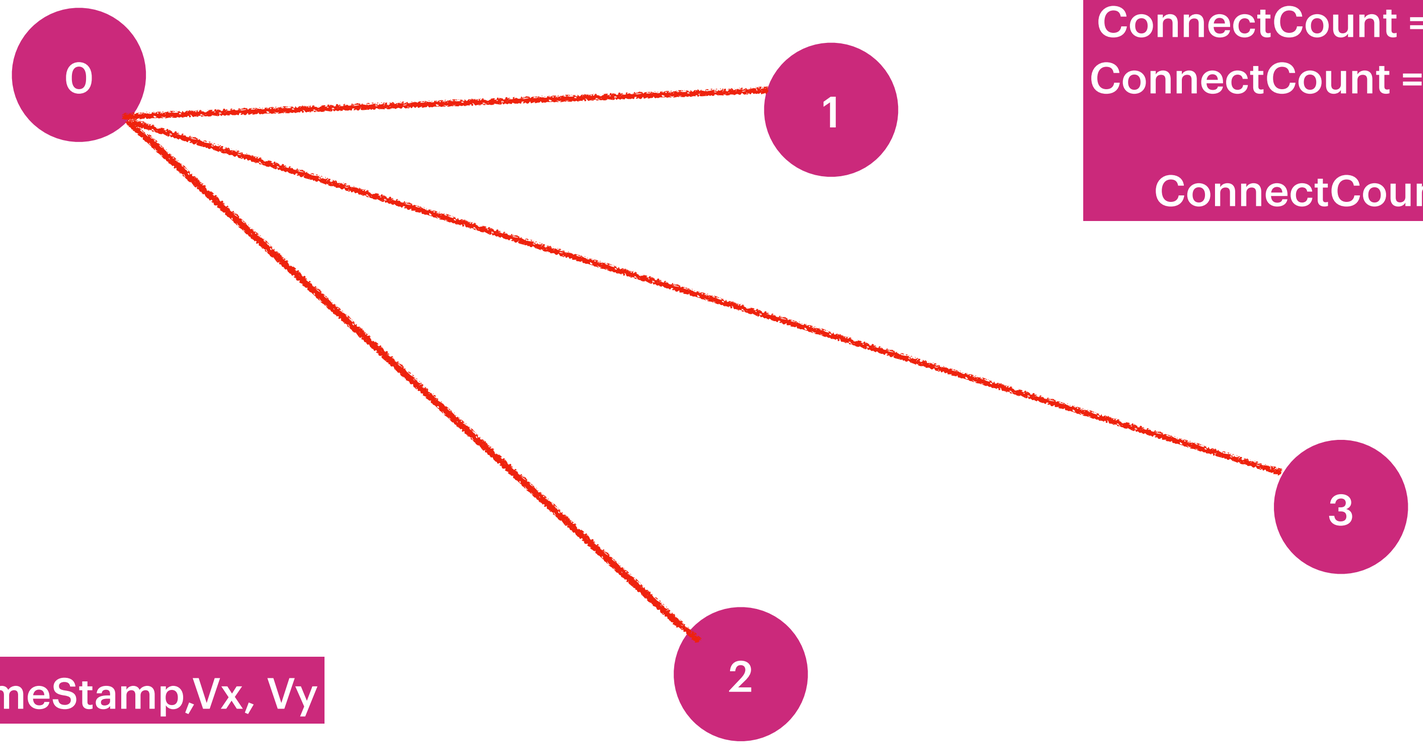
The Earliest Moment When Everyone Become Friends in Instagram

There are n people in a social group labeled from 0 to n - 1.
You are given an array logs where logs[i] = [timestampi, xi, yi] indicates that xi and yi will be friends at the time timestampi.
Friendship is symmetric. That means if a is friends with b, then b is friends with a. Also, person a is acquainted with a person b if a is friends with b, or a is a friend of someone acquainted with b.
Return the earliest time for which every person became acquainted with every other person. If there is no such earliest time, return -1.

Input: logs = [[20190101,0,1],[20190104,3,4],[20190107,2,3],[20190211,1,5],[20190224,2,4],[20190301,0,3],[20190312,1,2],[20190322,4,5]], n = 6
Output: 20190301

Input: logs = [[0,2,0],[1,0,1],[3,0,3],[4,1,2],[7,3,1]], n = 4
Output: 3

[[9,3,0],[0,2,1],[8,0,1],[1,3,2],[2,2,0],[3,3,1]]
4

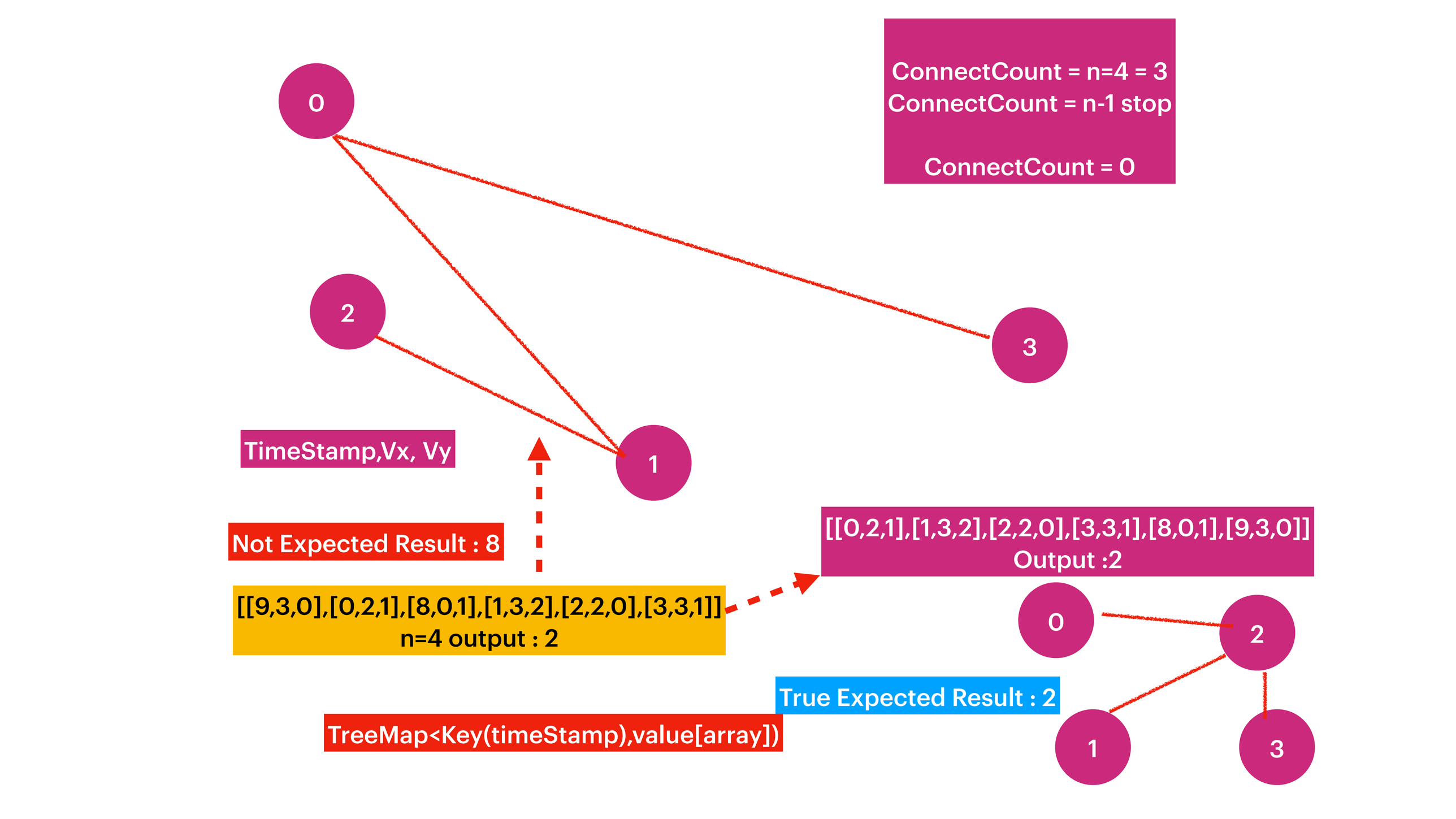


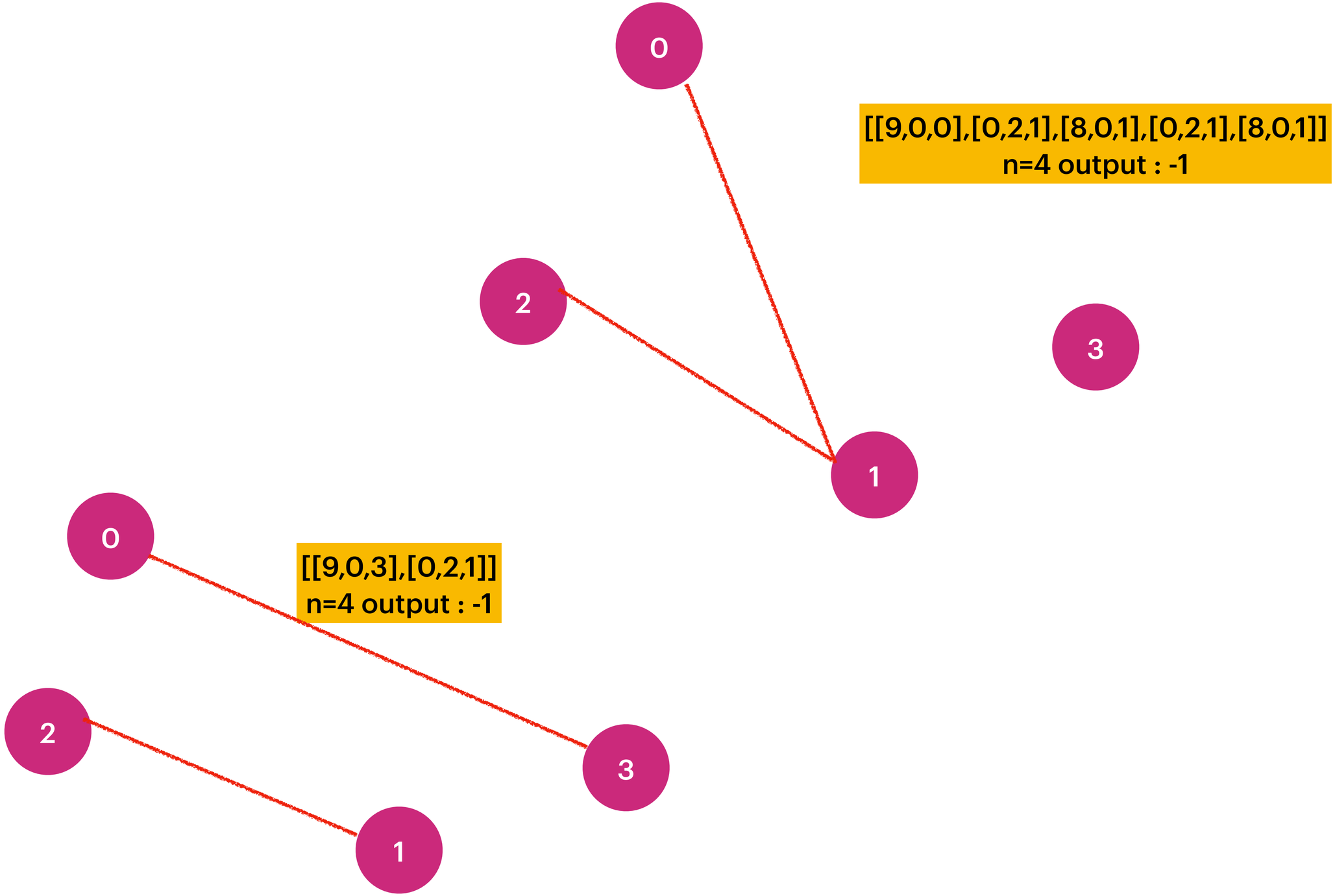
ConnectCount = n=4 = 3
ConnectCount = n-1 stop

ConnectCount = 0

TimeStamp,Vx, Vy

Input: logs = [[0,2,0],[1,0,1],[3,0,3],[4,1,2],[7,3,1]], n = 4
Output: 3


ConnectCount = n=4 = 3
ConnectCount = n-1 stop
ConnectCount = 0
0
2
3
1
TimeStamp,Vx, Vy
Not Expected Result : 8
[[9,3,0],[0,2,1],[8,0,1],[1,3,2],[2,2,0],[3,3,1]]
n=4 output : 2
[[0,2,1],[1,3,2],[2,2,0],[3,3,1],[8,0,1],[9,3,0]]
Output :2
0
2
1
3
True Expected Result : 2
TreeMap<Key(timeStamp),value[array])


0
2
3
1
[[9,0,0],[0,2,1],[8,0,1],[0,2,1],[8,0,1]]
n=4 output : -1
0
2
3
1
[[9,0,3],[0,2,1]]
n=4 output : -1