We need to find out whether a tree is a tree or not, given a graph at the beginning

For it to be a tree, it must not have a loop, that is, we need to know whether the tree has a loop or not

We have a graph and the loop should not be connected to each other, for which we will open a method

Using this method, we can find out if it has a loop in the tree

As you can see, it starts at 0 and goes to 1, then it goes to 2. Then it goes back to 2 to 0, and the loop is found.

We have a non-tree loop in the second graph because 0 1 2 is connected to each other

We open an object called a graph and we have 5 elements and then enter it using addEdge

0 we have 1 connected

we have a method called addEdge

We go to the first graph class and we also have int V private

Our array adj is inside LinkedList and can only contain numbers because it is integer

Let's go to the Graph constructor and say that V is equal to v

We call 0 to 5 in our loop and 0 of our array is equal to LinkedList

then we converted the array to a number in the Graph constructor, and in the addEdge method, for example, we replaced 0 with 1.

and in the bollean isCyclicUtil method, our array returns true

We entered Integer i with while and if element i of our visited array fails, this if program executes and goes to the isCyclicUtil method and returns true.

If it is not equal to parent, it returns true, and if none of these is true, it returns false

and using the boolean isTree method, we equalize the visited array v and go with the for program, and visisted returns us false, and if we put 3 variables in the isCyclicutil method, it returns false.

then go through the for program again until v returns false if visited [u]