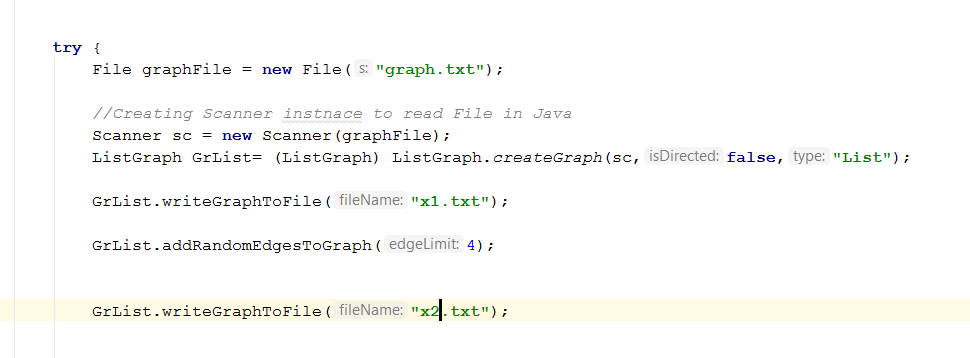
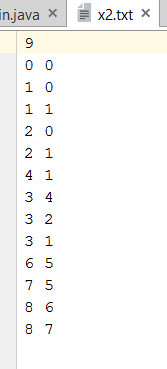
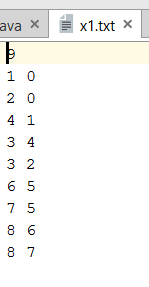
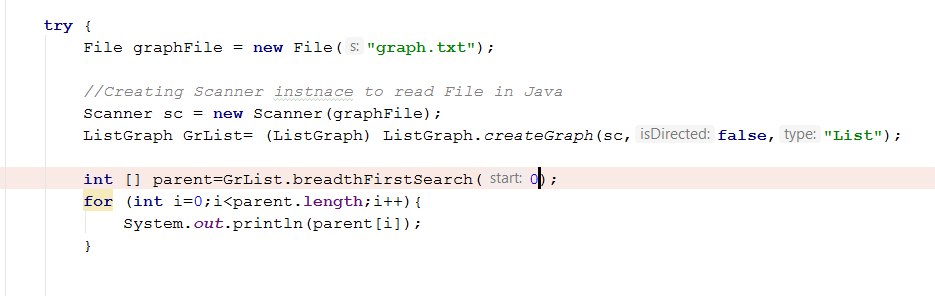
addRandomEdgesToGraph method test case.

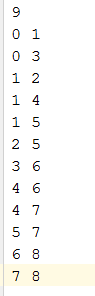


addRandomEdgesToGraph method test case result.

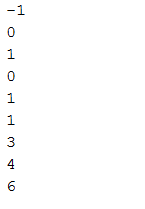


breadthFirstSearch method test case.

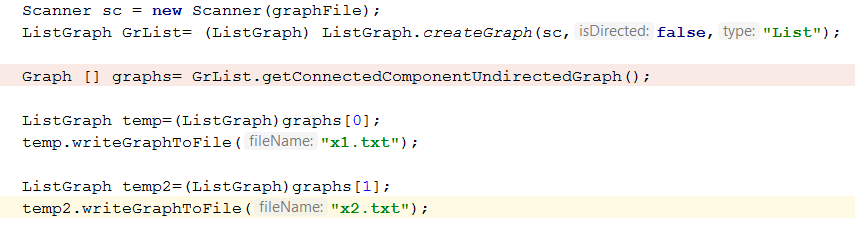
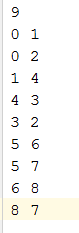




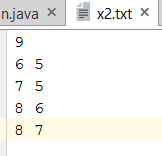
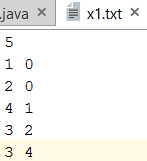
breadthFirstSearch method test case result.



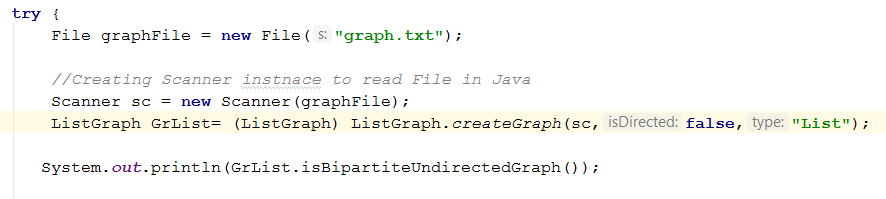
getConnectedComponentUndirectedGraph test case.



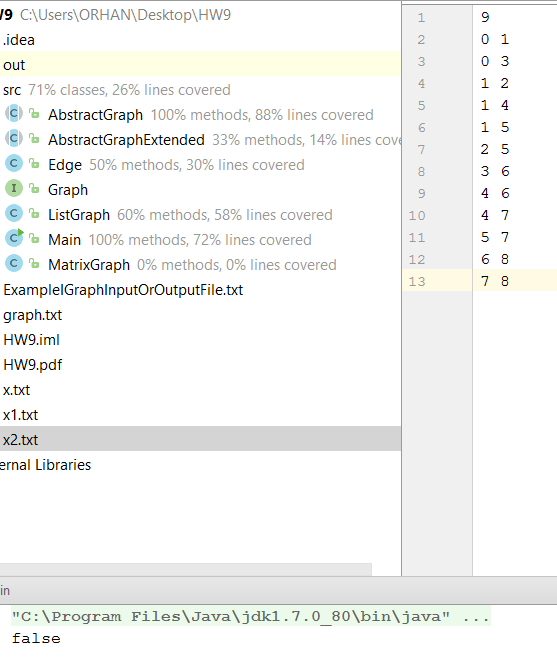
getConnectedComponentUndirectedGraph test case results.



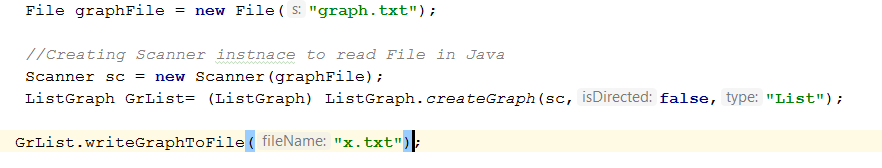
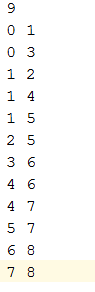
isBipartiteUndirectedGraph test cases and results.



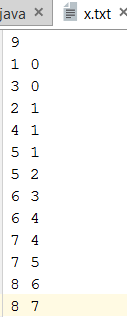




writeGraphToFile testcase.



Result.



Problem Solution Approach

addRandomEdgesToGraph

I have used isEdge and insert methods abstractGraph class and Random function of java.

isEdge to check if edge exist and if not insert new edge.7

breadthFirstSearch

I have implemented the algorithm written in the book.

getConnectedComponentUndirectedGraph

I have used an algorithm similar to breadthfirst search.Breadhtfirst search cannot travel all nodes if some group of nodes are isolated.

So when the algorithm breaks out of while(!theQueue.isEmpty()) i have started a for loop and checked if all the numbers from 0 to getnumV() has been identified.When we find a number that hasn’t been identified yet ,we start breadthfirstSearch algorithm again.

isBipartiteUndirectedGraph

I have used an int array called color.I have applied the same algorith from getConnectedComponentUndirectedGraph.

İf the current edge’s source color is 0 their neighbours color is 1.

İf the current edge’s source color is 1 their neighbours color is 0.

İf one of current edge’s neigbours has been assigned a color and it is same as the current edge’s source color ,it means the graph is not bipartite.

writeGraphToFile

i use breadthfirstsearch algorithm and store every edge source and destination in and arraylist.Then in every loop i check if the arraylist contains the same edge in reverse to avoid printing same edge twice.

i