

1)

- a) We want to find the in-degrees and print the adjacency list.
- 1) loop through the in-degrees array to fill it with 0s
 - 2) loop through the in-degrees array to get the in-degree of each node
 - while iterator has the next value
 - increment the in-degree's next iteration
 - 3) loop through the length of the in-degree's array so long as there are more than 2 nodes
 - loop through the in-degrees array
 - if the index of the in-degree array is 0
 - decrement the length of the in-degrees array
 - set indexTracker to the current index
 - set the current index to the in-degree array
 - if the indexTracker wasn't set (-1)
 - return true
 - else
 - set the in-degree array at the current indexTracker to -1
 - loop through adjacent matrix
 - if adjacency matrix's indexTracker and index are set to 1
 - decrement the in-degrees
- b) for (i = 0; i < in_degree.length; i++) // n
- Output:** in_degree [i] = 0;
- for (i = 0; i < in_degree.length; i++) // n
- iterator = adjList.get(i).iterator()
- while (iterator.hasNext()) // 1
- in_degree[(int)iterator.next()]++
- while (in_degree_length > 2) // 1
- for ((i = 0; i < in_degree.length; i++) // n
- if (in_degree[i] == 0) // 1
- in_degree_length--
- iTracker = i
- i = in_degree.length
- if (iTracker == -1) // 1
- Output:** return true
- else
- for (j = 0; j < adjMat[iTracker].length; j++) // n
- if (adjMat[iTracker][j] == 1) // 1
- in_degree[j] --

c) $O(n) = (V + E^2)$