***CITS2200 Project Report***

***By Adrian Bedford 22973676 & Siwei Lin 22967534***

allDevicesConnected:

This method uses a breadth-first search, visiting each node in adjlist from level to level. A queue is initialised and is given the 0th node to hold at the beginning of the algorithm. So long as the queue is not empty, that the nodes already visited have children, the loop will continue to run, iterating through the length of adjlist and looking at what nodes a parent is connected to. The array ‘return’ holds a sequence of Boolean values where each index, as representative of a node, denotes if it has been visited or not. When the BFS reaches a child it has not yet seen from inside the for loop, it updates the respective value inside ‘return’ to true and adds the node to the queue. From here it now uses the latest found node and checks what children it has, following the same process down adjlist until there are no more connected nodes. At this stage, the BFS terminates since we remove the head of the queue each time we finish the process of finding a new child, and as there are no new children we have reached the end of the graph. The ‘results’ array will now contain an accurate list of nodes that have either been or not been visited. We now iterate over ‘results’ to update the value of a variable ‘connected’, which will be false if any node has not been visited or true if all nodes have been visited. ‘connected’ is then returned as the answer to method 1.

**IDK MAYBE NOT RIGHT**

The time complexity of method 1 is O(D + L), equivalent to O(N). At worst case the while loop won’t break until every single node has been added to queue and visited by the BFS, so already we must process the entire length of adjlist. Inside while there is a for loop that in any case will iterate through the entire length of adjlist, and will also need to . Nested within that is a conditional that in all cases will be accessing ‘results’ for the length of adjlist. Finally, there is a for loop at the bottom of the method…

numPaths:

closestInSubnet

* Argument for correct answer
* Time complexity analysis

maxDownloadSpeed:

* Argument for correct answer
* Time complexity analysis

***Bibliography***

a