The reason why my algorithm is slow is because I use a lot of nested loops when writing the BFS() method and I'm calling BFS method with 2 for loop. In the first loop, I check if the queue (which is named queue) I have added the vertices I visited is empty. In the second loop (the for loop inside the unvisitedNeighbor method) I find the unvisited vertices of my current vertex and return the index of that vertex. In the third loop, I'm checking to see if my current vertex has any unvisited vertices. If there is, I am updating the previously unvisited vertex in the previous array as visited. If the vertex (neighbour) I just visited is equal to the endvertex, I clear the queue and exit the loop. Then I travel the vertices in the previous array with the trace_route() method and create a route. I calculate the betweenness and closeness values of the elements in the route I created. So there is 5 nested loop. My code runs in about 15 minutes as seen in the image below.