I chose to implement my Breadth First Search function by using a while loop that repeats as long as the queue of vertices is still full. The code starts with just the starting node in the queue. This node is added to the visited set, I used a set so repeat entries could not be added, and it is added to the order list if it has not been visited. Then the list of adjacent vertices is taken from the provided adjacent_vertices() function. The adjacent vertices are checked if they have been visited and added to the queue. Then the next vertex in the queue is chosen. This repeats until the queue is empty.

The largest issue I had while developing this code was a hashing issue with the VertexEL elements. I was trying to see if VertexEL elements were in the visited set. This required the object to be hashed. I fixed this by adding the name of the vertices to the visited list themselves. The names of the Vertices could be checked against the visited set.

The function is in Big(O) complexity as required by the assignment.