Task - 4 answer

The BFS and DFS time complexity depends on how we store the graph. There are two ways to store a graph (1) Adjacency matrix and (2) Adjacency list.

In adjacency matrix, there are V vertices, then (V*V) matrix will be used to store the graph. In BFS, if matrix is used, we end up going each and every nodes to if they are connected or not. Thus, in BFS we go through V^2 elements atleast once using matrix. Therefore, the time complexity using matrix for BFS is $O(|V|^2)$. Similarly, the time complexity for DFS using matrix is also $O(|V|^2)$.

Again, in ~~adjac~~ terms of adjacency list, we visit all the vertices and all the edges in both BFS and DFS. Thus, the time complexity for both BFS and DFS using adjacency list is $O(|V| + |E|)$. Here, $|V|$ is number of vertices and $|E|$ is number of edges.

BFS can be used in an unweighted graph or evenly weighted graph to find shortest path for single vertex as BFS reaches the node with the least number of edges from the source node. On the other

hand, DFS does not ~~guree~~ guarantee the shortest path as it may or may not traverse more edges to reach the target vertices from the source. Therefore, DFS ~~sometimes~~ traverse ~~less~~ nodes randomly, resulting in traversing more or less or equal nodes than BFS. Moreover, the time complexity is same for both BFS and DFS. and so it can't be directly said ~~that~~ which is faster. However, by ~~the~~ looking at the ~~given t~~ output of the given task 2 and 3 it can be said that ~~the~~ Barry reaches the victory road first than me as using DFS, the number of traversal nodes are less than the number of traversal nodes using BFS.