BFS

Adjacency list: In order to perform BFS put any vertex within the line and pop the queue, Choose the beginning vertex, investigate all its adjoining vertexes make them as visited and explore all the non visited vertices until the queues becomes empty.

Time complexity: $O(V+E)$

Adjacency matrix: In matrix, for every vertex we have to traverse all the vertices and check the vertices is visited or not.

Time Complexity: $O(V^2)$

DFS

Adjacency list: In the adjacency list, each node maintains a list of all its adjacent edges, for each node we discover all of its neighbours by traversing its adjacency list.
So time complexity of DFS is
$O(V) + O(E) = O(V+E)$

Adjacency matrix: If we use adjacency matrix, for each node we have to traverse an entire row of length V.
So, time complexity $= O(V*V)$
$= O(V^2)$

Here DFS algorithm will reach the victory road first because DFS algorithm is better when the destination is far while BFS is better when the destination is closure. On the other hand, in the task 2, we use BFS algorithm and in the output we notice that we have to traverse 9 nodes. Again, in the task 3 we use DFS algorithm, here we notice that in the output we have to traverse just 7 nodes which is less than BFS algorithm.