TaskO4

In BFS, for adjacency list, the total time complexity

$$T(n) = \underbrace{O(v)}_{\text{for each vertex}} + \underbrace{O(e)}_{\text{for each edge}} + \underbrace{O(1)}_{\text{single operations}} = O(v+e)$$

However, for adjacency matrix, the total time complexit

$$T(n) = \underbrace{O(1)}_{\text{single operations}} + \underbrace{O(v^2)}_{\text{checking rows and columns}}$$

$$= O(v^2)$$

In DFS, for adjacency list, the total time complexity

$$T(n) = \underbrace{O(1)}_{\text{single operations}} + \underbrace{O(v)}_{\text{for each vertex}} + \underbrace{O(e)}_{\text{for each edge}}$$

$$= O(v+e)$$

However, for adjacency matrix, the total time complexity -

$$T(n) = \underbrace{O(1)}_{\text{single operations}} + \underbrace{O(v2)}_{\text{checking for rows and columns.}}$$

Even though the time complexity of both the algorithms are same, however we can notice that in DFS, there are less nodes visited in order to reach the endpoint whereas in BFS, more nodes are visited to reach the endpoint. So, Gary gets victory as he reaches faster.