

Answer to the Question no: (3)

part ①

Given,

N places = vertex

M roads = edges

According to my code,

the while loop traverses through all vertex

\therefore time complexity = $O(V)$

Secondly, there are two inner for loop.

\therefore for first one, time complexity = $O(V)$

\therefore for second loop, time complexity = $O(V+E)$

\therefore inner time complexity = $O(V) + O(V+E) + O(1)$
 $= O(V) + O(V+E)$

\therefore Total = $O(V) (O(V) + O(V+E))$
 $= O(V^2)$

part ②

the algorithm is same as part ① and under the second for inner loop is the modified part, but it will not affect the time complexity because its time complexity = $O(1)$

Lastly, for the last while loop, time complexity = $O(E)$

$$\begin{aligned}\therefore \text{total} &= O(1) \cdot (O(V) + O(V+E) + O(E)) \\ &= O(V^2)\end{aligned}$$

Part ③

If the number of titans in all roads become 1, then we should not be concerned about the weight because BFS will give the shortest path.

\therefore The required algorithm is BFS whose time complexity will be $O(N+M)$