For the given weighted graphs we used the algorithm of djkstra to find single source shortest path

Time complexity of my provided solution for problem 1 and 2 :

N = total vertex, M = total edge

① Initializing all data structures = $O(N)$

② while queue not empty : - $O(N)$
                                       [since N vertex has to be removed]

    ③ Finding minimum — $O(N+N) = O(N^2)$

    ④ visiting adjecent vertexes - $O(N * N-1)$
                                       $= O(N^2)$

$\therefore$ Time complexity $O(N) + O(N^2) + O(N^2) = O(N^2)$

In Problem 2:
we sorted two arrays in nlogn time with .sort()
But the time complexity remains $O(N^2)$
as the rest of the solve is same as problem 1

However, in problem # 2 b, if we consider the cases, the time complexity it will be increased to $O(cases * N)$.

If the numbers of titans in each node is exactly 1, then it is an unweighted graph. In that case we can solve it with BFS algorithm as its time complexity is $O(N+M)$ for

Inputs given, will be $(V(\alpha)$ Graph, (S) source, Destination$)$