(4)

from task 2,

There are N places (vertex)
There are M roads (edges)

Time complexcity : $O(M \log N)$

for task 3,

we need more memory for task 3 than task 2. But the time complexcity is same. number of places visited in shortest path = p

time complexcity : $O(p)$

Dijkstra time complexcity = $O(M \log N)$

∴ Total time complexcity = $O(M \log N + p)$

if the number of titans in every road is exactly 1, then we can find the shortest path using BFS. It is a $O(M+N)$ algorithm. This algorithm gives us the least needed roads.