

Answer to task 3:

Time complexity for given problem:

function Dijkstra (Graph, source):

dist[source] \leftarrow 0

Create Queue

create array for visited node

for each vertex v in graph:

build
heap
 $O(V)$

if $v \neq \text{source}$

dist[v] $\leftarrow \infty$

pre[v] $\leftarrow \text{Null}$

add v to Q with priority value

visited[v] $\leftarrow \text{false}$

while Q is not empty:

$O(V \log V)$ $\rightarrow u \leftarrow Q.\text{extract_min}()$

if visited[u]:

continue

visited[u] $\leftarrow \text{True}$

for each neighbour v of u :

if dist[u] + length(u, v) < dist[v]:

dist[v] = dist[u] + length(u, v)

prev[v] $\leftarrow u$

$O(E \log V)$

\rightarrow add v to Q with priority value[v]

∴ Total time complexity is

$$O(V) + O(V \log V) + O(E \log V)$$

$$= O((V+E) \log V)$$

#

If the number of titans of each roads is exactly one, that will be a graph of no weight.

BFS algorithm is used to find path

if weight is similar or no weight.
inputs:

1	0			1	2	1
2	1			2	3	1
1	2	1		1	4	1
5	6			4	3	1
3	5	1		2	5	1