

Name: Mollah Md. Saif

Lab Assignment-04

ID no. 20101416

Course : CSE 221

Section : 10

Task-3

For task-1,

The algorithm used is traversing through the neighbours by ~~put~~ a queue. Here, this code is completely similar to BFS with addition of few lines of code. The main difference is in the queue implementation. Here, we use binary heap to form a min-heap-priority-queue which ~~are~~ have a time complexity of $\log(n)$ for enqueueing and dequeueing.

$$\text{BFS complexity} = O(V+E)$$

\therefore We are ~~enqueue~~ enqueue and the unvisited vertex only and also dequeue the.

$$\begin{aligned}\therefore \text{Total time complexity of queue} &= O(2 \log V) \\ &= O(\log(V))\end{aligned}$$

$$\begin{aligned}\therefore \text{Final time complexity of the whole code} \\ &= O((V+E) \log(V))\end{aligned}$$

For task-2,

the time complexity of ~~it~~ is same as task-1 since there is only one extra operation "prev" which is $O(1)$.

If the number of titans were 1 in every road, then the graph would become an unweighted graph. So, no priority-~~que~~ queue were needed and we would use normal queue instead which have a time complexity of $O(n)$.

Here, the algorithm is completely same as bfs code.
 ~~\therefore The total time complexity~~
algorithm for finding shortest path.

\therefore The time complexity would become $O(V+E)$