Pseudocode

```
def network(times, n, k):
    instantiate priority queue heap and push the first node to it              → O(1)
    define adjacency list with all of our nodes                               → O(E)
    declare set of visited nodes
    while ( heap is not empty):                                               → O(V)
        get current node and its time from the priority queue                 → O(logv)

        if node is not in visited:                                            → O(1)
            add node to visited set                                           → O(1)
            if visited list is equal to total # nodes, return current node's time → O(1)

            for all neighbors of the current node:                            → O(Elogv)
              push(current node time + neighbor_time, neighbor) to heap
    return -1
```

Time complexity

Time complexity: O(ElogV) + O(VlogV) since but since there are more edges the total time complexity of this problem would be O(ElogV)