743. Network Delay Time

743. Network Delay Time

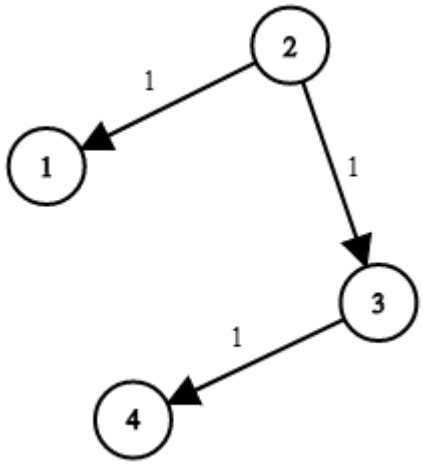
Medium

2930255Add to ListShare

You are given a network of `n` nodes, labeled from `1` to `n`. You are also given `times`, a list of travel times as directed edges `times[i] = (ui, vi, wi)`, where `ui` is the source node, `vi` is the target node, and `wi` is the time it takes for a signal to travel from source to target.

We will send a signal from a given node `k`. Return the time it takes for all the `n` nodes to receive the signal. If it is impossible for all the `n` nodes to receive the signal, return `-1`.

Example 1:




```
Input: times = [[2,1,1],[2,3,1],[3,4,1]], n = 4, k = 2
Output: 2
```

Example 2:

```
Input: times = [[1,2,1]], n = 2, k = 1
Output: 1
```

Example 3:

```
Input: times = [[1,2,1]], n = 2, k = 2
Output: -1
```

```
import heapq
class Solution:
    def networkDelayTime(self, times: List[List[int]], n: int, k: int) ->
int:
        graph = defaultdict(list)
        for i in range(len(times)):
            u,v,w = times[i]
            graph[u].append([v,w])

        visited = [False]*(n+1)
        queue = []
        heapq.heappush(queue,(0,k))
        cost = []
        # print(queue)
        while len(queue):
            wt,node = heapq.heappop(queue)
            if visited[node]==True:
                continue
            cost.append(wt)
            visited[node] = True
            for nbr in graph[node]:
                tempNode,weight = nbr
                if visited[tempNode]==False:
                    heapq.heappush(queue,(wt+weight,tempNode))
        return max(cost) if len(cost)==n else -1
```