

Problem 743: Network Delay Time

Problem Information

Difficulty: Medium

Acceptance Rate: 58.99%

Paid Only: No

Tags: Depth-First Search, Breadth-First Search, Graph, Heap (Priority Queue), Shortest Path

Problem Description

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 $times[i] = (u_i, v_i, w_i)$, where u_i is the source node, v_i is the target node, and w_i 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 **minimum** 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

Constraints:

$1 \leq k \leq n \leq 100$ $1 \leq times.length \leq 6000$ $times[i].length == 3$ $1 \leq u_i, v_i \leq n$
 $u_i \neq v_i$ $0 \leq w_i \leq 100$ All the pairs (u_i, v_i) are **unique**. (i.e., no multiple edges.)

Code Snippets

C++:

```
class Solution {  
public:  
    int networkDelayTime(vector<vector<int>>& times, int n, int k) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int networkDelayTime(int[][] times, int n, int k) {  
  
    }  
}
```

Python3:

```
class Solution:  
    def networkDelayTime(self, times: List[List[int]], n: int, k: int) -> int:
```