Problem 815: Bus Routes

Problem Information

Difficulty: Hard
Acceptance Rate: 46.99%
Paid Only: No
Tags: Array, Hash Table, Breadth-First Search

Problem Description

You are given an array `routes` representing bus routes where `routes[i]` is a bus route that the `ith` bus repeats forever.

* For example, if `routes[0] = [1, 5, 7]`, this means that the `0th` bus travels in the sequence `1 -> 5 -> 7 -> 1 -> 5 -> 7 -> 1 -> ...` forever.

You will start at the bus stop `source` (You are not on any bus initially), and you want to go to the bus stop `target`. You can travel between bus stops by buses only.

Return _the least number of buses you must take to travel from_`source` _to_`target`. Return `-1` if it is not possible.

Example 1:

Input: routes = [[1,2,7],[3,6,7]], source = 1, target = 6 **Output:** 2 **Explanation:** The best strategy is take the first bus to the bus stop 7, then take the second bus to the bus stop 6.

Example 2:

Input: routes = [[7,12],[4,5,15],[6],[15,19],[9,12,13]], source = 15, target = 12 **Output:** -1

Constraints:

* `1 <= routes.length <= 500`. * `1 <= routes[i].length <= 105` * All the values of `routes[i]` are **unique**. * `sum(routes[i].length) <= 105` * `0 <= routes[i][j] < 106` * `0 <= source, target < 106`

Code Snippets

C++:

```
class Solution {
public:
int numBusesToDestination(vector<vector<int>>& routes, int source, int
target) {

}
};
```

Java:

```
class Solution {
public int numBusesToDestination(int[][] routes, int source, int target) {

}
}
```

Python3:

```
class Solution:
def numBusesToDestination(self, routes: List[List[int]], source: int, target:
int) -> int:
```