815. Bus Routes

DescriptionHintsSubmissionsDiscussSolution

We have a list of bus routes. Each `routes[i]` is a bus route that the i-th bus repeats forever. For example if `routes[0] = [1, 5, 7]`, this means that the first bus (0-th indexed) travels in the sequence 1->5->7->1->5->7->1->... forever.

We start at bus stop `S` (initially not on a bus), and we want to go to bus stop `T`. Travelling by buses only, what is the least number of buses we must take to reach our destination? Return -1 if it is not possible.

```
Example:
Input:
routes = [[1, 2, 7], [3, 6, 7]]
S = 1
T = 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.
```

Note:

- `1 <= routes.length <= 500`.
- `1 <= routes[i].length <= 500`.
- `0 <= routes[i][j] < 10 ^ 6`.

Seen this question in a real interview before?

- Difficulty:Hard
- Total Accepted:1.2K
- Total Submissions:4K
- Contributor:awice
-
- Subscribe to see which companies asked this question.

  Related Topics BFS

-

```
int numBusesToDestination(vector<vector<int>> &routes, int S, int T)
{
     if(S==T) return 0;
     int n = routes.size();
     vector<unordered_set<int>> bus(n,unordered_set<int>());
     for(int i=0;i<(int)routes.size();++i)
     {
          vector<int> route = routes[i];
          for(auto elem:route)
          {
               bus[i].insert(elem);
          }
     }
     queue<unordered_set<int>> q;
     vector<bool> visited(bus.size(),false);
     for(int i=0;i<(int)bus.size();++i)
     {
          if(bus[i].count(S))
          {
               q.push(bus[i]);
               visited[i]=true;
          }
     }
     int res = 1;
     while(!q.empty())
     {
          int n = q.size();
          cout << n << endl;
          for(int i=0;i<n;++i)
          {
               unordered_set<int> cur = q.front();
               q.pop();
               unordered_set<int>::iterator it;
               for(it = cur.begin();it!=cur.end();++it)
               {
                    auto e = *it;
                    if(e==T) return res;
                    for(int j=0;j<(int)bus.size();j++)
                    {
                         if(!visited[j] && bus[j].count(e))
                         {
                              visited[j]=true;
                              q.push(bus[j]);
                         }
```

```
                    }
                }
            }
            res++;
        }
        return res;
    }
```