Rail

Solution

Way : Ad Hoc
Query complexity: 3(N-1)
Time complexity : O(N log N)
** This problem can be solved by the following steps :

First, we know that station 0 is C type, and its location. We can query all the other stations' distances from station 0, we call this : `dis[0][i]`

Second, we sort all the `dis[0][i]`, and obviously, the station x with the shortest distance `dis[0][x]` must be the first D type location after station 0.

Third, we process each station one by one according to their shortest distance with station 0 (that is, the order obtained in second step). For each station processed, we determine its type and location immediately as follows:

3.1 First, we maintain the information `(location,id)` of the leftmost C type and the rightmost D type as the algorithm proceeds.

3.2 To process the current station k, we use two queries, `query(k,leftmost C type)` as `dis[k][L]`, `query(k, rightmost D type)` as `dis[k][R]`. And we also have `dis[0][k]`. By some observations, we know that either `dis[k][L]` or `dis[k][R]` is achieved with a 'direct' route (without moving forth and back).

For example, we have only 4 cases to consider :

```
a. dis[k][L] is a direct route
       (   )   )
       L   k   R

b. dis[k][L] is a direct route
       (       )    )
       L       R    k

c. dis[k][R] is a direct route
    (  (       )
    k  L       R

d. dis[k][R] is a direct route
       (   (   )
       L   k   R
```

By careful analysis (some if/else conditions), we can get the answer. Sometimes, we may need extra information to resolve the four cases, where we can check with `dis[0][k]`.

Take case (a) for example :

```
a. dis[k][L] is a direct route
       (   )   )
       L   k   R
```

the location of k might be L + `dis[k][L]`, Then use this location, we need to check if `dis[k][R]` is reasonable or not.

```
a.1
       ( ( )   )
       L p k   R
```

There might be some p(type C) between L and k, So the `dis[k][R]` is `dis[p][R]+dis[p][k]`, we need to check whether p exists or not. If p doesn't exist, then case (a) might be wrong, then try cases (b),(c), and (d) until we find the answer.