



HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.



ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Applied Algorithm Lab

Telco data check & query

ONE LOVE. ONE FUTURE.

- Perform some queries on a telco data
- Input: The first block of data:

call <from_number> <to_number> <date> <from_time> <end_time>

<from_number> and <to_number> are string of 10 characters (a phone number is correct if it contains only digits 0,1,...,9, otherwise, the phone number is incorrect)

- Queries: The second block of data: 4 types of queries
 - ?check_phone_number: print to stdout (in a new line) value 1 if no phone number is incorrect
 - ?number_calls_from <phone_number>: print to stdout (in a new line) the number of times a call is made from <phone_number>
 - ?number_total_calls: print to stdout (in a new line) the total number of calls of the data
 - ?count_time_calls_from <phone_number>: print to stdout (in a new line) the total time duration (in seconds) the calls are made from <phone_number>

• Example

stdin	stdout
call 0912345678 0132465789 2022-07-12 10:30:23 10:32:00	1
call 0912345678 0945324545 2022-07-13 11:30:10 11:35:11	2
call 0132465789 0945324545 2022-07-13 11:30:23 11:32:23	4
call 0945324545 0912345678 2022-07-13 07:30:23 07:48:30	398
#	120
?check_phone_number	
?number_calls_from 0912345678	
?number_total_calls	
?count_time_calls_from 0912345678	
?count_time_calls_from 0132465789	
#	

- Idea to solve
 - ?check_phone_number and @number_total_calls:
 - easy, use 1 variable to store
 - write a function to check the phone is valid or not
 - ?number_calls_from <phone_number>
 - ?count_time_calls_from <phone_number>: 2 ways to deal:
 - use 2 mappings: map <string,int> numberCalls, timeCall;
 - use a binary search tree

- use binary search tree: programming

```
struct Node{  
    string from;  
    int duration; //second  
    int nbCall; //number of Call  
    Node *right_child;  
    Node *left_child;  
};
```

- functions:

```
Node *makeNode(string from1, int duration1, int nbCall1)
```

```
Node *find(string number, Node *r)
```

```
Node *insert(string from, int duration, int nbCall, Node *r)
```




HUST

THANK YOU !