

# Thuật toán ứng dụng thực hành

Buổi 1

- Telco Data Check & Analyze
- MAZE
- Range Minimum Query
- Largest Black SubRectangle

# 1.Telco Data Check & Analyze (Kiểm tra và phân tích dữ liệu log cuộc gọi thoại)

---

- Một nhà mạng muốn thực hiện truy vấn dữ liệu log lịch sử cuộc gọi trong ngày, dữ liệu log này được format dạng:

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

Ý nghĩa của các trường là

- Từ khóa `call`: đây là log cuộc gọi điện thoại
- `<from_number>` và `<to_number>`: là SDT gọi và nhận cuộc gọi, là kiểu xâu ký tự độ dài 10 (chỉ gồm các chữ số 0-9)
- `<date>`: Là ngày thực hiện cuộc gọi theo định dạng YYYY-MM-DD (VD. 2022-10-21)
- 

Chú ý:

- Số lượng log cuộc gọi này có thể lớn tới 100000 dòng
- Các tham số ngăn cách với nhau bởi 1 dấu cách trống

## 1.Telco Data Check & Analyze (Kiểm tra và phân tích dữ liệu log cuộc gọi thoại)

---

Các truy vấn dữ liệu log được đưa vào với định dạng bắt đầu bằng dấu ?, trong đó

- **?check\_phone\_number**: in ra màn hình (dòng mới) giá trị 1 nếu các số điện thoại đều hợp lệ
- **?number\_calls\_from <phone\_number>**: in ra màn hình (dòng mới) số cuộc gọi được xuất phát từ SDT <phone\_number>
- **?number\_total\_calls**: in ra màn hình (dòng mới) tổng số cuộc gọi có trong log
- **?count\_time\_calls\_from <phone\_number>**: in ra màn hình (dòng mới) tổng thời gian gọi (tính theo second) xuất phát từ SDT <phone\_number>

Chú ý:

- Số lượng truy vấn cũng có thể lên tới 100000 dòng
- Các tham số ngăn cách với nhau bởi 1 dấu cách trống

# 1.Telco Data Check & Analyze

stdin	Stdout
call 0912345678 0132462157 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	
#	
?check_phone_number	
?number_calls_from 0912345678	
?number_total_calls	

## 2.MAZE

---

- Một mê cung hình chữ nhật được biểu diễn bởi 0-1 ma trận  $N \times M$  trong đó
  - $A[i,j] = 1$  thể hiện ô  $(i,j)$  là tường gạch và
  - $A[i,j] = 0$  thể hiện ô  $(i,j)$  là ô trống, có thể di chuyển vào.
- Từ 1 ô trống, ta có thể di chuyển sang 1 trong 4 ô lân cận (lên trên, xuống dưới, sang trái, sang phải) nếu ô đó là ô trống.
- Xuất phát từ 1 ô trống trong mê cung, hãy tìm đường ngắn nhất thoát ra khỏi mê cung.
- **Input**
  - Dòng 1: ghi 4 số nguyên dương  $n, m, r, c$  trong đó  $n$  và  $m$  tương ứng là số hàng và cột của ma trận  $A$  ( $1 \leq n, m \leq 999$ ) và  $r, c$  tương ứng là chỉ số hàng, cột của ô xuất phát.
  - Dòng  $i+1$  ( $i=1, \dots, n$ ): ghi dòng thứ  $i$  của ma trận  $A$
- **Output**
  - Ghi giá số bước cần di chuyển ngắn nhất để thoát ra khỏi mê cung, hoặc ghi giá trị -1 nếu không tìm thấy đường đi nào thoát ra khỏi mê cung.

## 2.MAZE

- Example

stdin	stdout
8 12 5 6 1 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 1 0 1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 1 0 1 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 1 0 1 1 1 0 1 0 1	7

### 3.Range Minimum Query (Giá trị nhỏ nhất trong khoảng)

---

- Cho 1 đoạn gồm  $n$  số nguyên với giá trị  $a_0, \dots, a_{n-1}$ , ta định nghĩa  $rmq(i, j)$  là giá trị nhỏ nhất trong đoạn từ  $a_i$  tới  $a_j$  (giá trị số nhỏ nhất trong các số  $a_i, a_{i+1}, \dots, a_j$ ).
- Ví dụ dãy 10 phần tử 1,5,3,7,8,43,23,5,12,7 thì

$$rmq(0,9) = 1$$

$$rmq(1,9) = 3$$

$$rmq(3,5) = 7$$

- Với đầu vào là  $m$  đoạn  $(i_1, j_1), \dots, (i_m, j_m)$ , giá trị tổng của các  $rmq$  định nghĩa trên  $m$  cặp được tính như sau  $Q = rmq(i_1, j_1) + \dots + rmq(i_m, j_m)$
- Input
  - Dòng 1: là số nguyên  $n$  ( $1 \leq n \leq 10^6$ )
  - Dòng 2: chứa giá trị các phần tử trong đoạn ban đầu  $a_0, \dots, a_{n-1}$  ( $1 \leq a_i \leq 10^6$ )
  - Dòng 3: là giá trị  $m$  ( $1 \leq m \leq 10^6$ )
  - Các dòng tiếp theo từ  $k+3$  ( $k = 1, \dots, m$ ): là các cặp giá trị  $i_k, j_k$  ( $0 \leq i_k < j_k < n$ )
- Output: in ra giá trị  $Q$



### 3.Range Minimum Query

- Example

stdin	stdout
16 2 4 6 1 6 8 7 3 3 5 8 9 1 2 6 4 4 1 5 0 9 1 15 6 10	6

## 4. Largest Black SubRectangle

---

- Một hình chữ nhật kích thước  $n \times m$  được chia thành các ô vuông con  $1 \times 1$  với 2 màu đen hoặc trắng. Hình chữ nhật được biểu diễn bởi ma trận  $A(n \times m)$  trong đó  $A(i, j) = 1$  có nghĩa ô hàng  $i$ , cột  $j$  là ô đen và  $A(i, j) = 0$  có nghĩa ô vuông hàng  $i$  cột  $j$  là ô trắng.
- Hãy xác định hình chữ nhật con của bảng đã cho bao gồm toàn ô đen và có diện tích lớn nhất.
- **Dữ liệu**
  - Dòng 1: chứa số nguyên dương  $n$  và  $m$  ( $1 \leq n, m \leq 1000$ )
  - Dòng  $i+1$  ( $i = 1, \dots, n$ ): chứa hàng thứ  $i$  của ma trận  $A$
- **Kết quả**
  - Ghi ra diện tích của hình chữ nhật lớn nhất tìm được

## 4.Largest Black SubRectangle

---

- Example

stdin	stdout
4 4 0 1 1 1 1 1 1 0 1 1 0 0 1 1 1 0	6