

# Vitaly and Strings

Big-O Blue - Lecture 1: Dynamic Array & String

# Tóm tắt đề bài

# Tóm tắt đề bài

Vitaly có hai chuỗi cùng độ dài là **S** và **T** chỉ bao gồm các chữ cái **tiếng Anh viết thường**. Biết chuỗi **S** có thứ tự từ điển nhỏ hơn chuỗi **T**.

Hãy tìm chuỗi nào được xếp theo thứ tự từ điển, **lớn hơn** chuỗi **S** và **nhỏ hơn** chuỗi **T**. Đây là chuỗi cũng chứa các chữ cái tiếng Anh viết thường và có độ dài bằng độ dài của chuỗi **S** và **T**.

# Mô tả Input/Output

**Input:** bao gồm 1 test.

Dòng đầu tiên chứa chuỗi ***S*** ( $1 \leq \text{độ dài } S \leq 100$ ) gồm các chữ cái tiếng Anh viết thường.

Dòng thứ hai chứa chuỗi ***T*** (độ dài ***S*** = độ dài ***T***) gồm các chữ cái tiếng Anh viết thường.

**Output:**

Nếu tồn tại chuỗi **hợp lệ** thì in ra (nhiều đáp án chỉ cần in 1 cái). Ngược lại thì in ra “No such string”.

# Giải thích ví dụ

# Ví dụ 1

**Input:**

S = “k”

T = “m”

**Output:**

“l”

**Lí do:**

Dễ thấy “k” < “l” < “m”, trong bảng chữ cái.

# Ví dụ 2

## Input:

S = "klmnopq"

T = "klmpopq"

## Output:

"klmnopr"

## Lí do:

So với S, ta có:

→ S = "klmnop<sub>q</sub>" < "klmnop<sub>r</sub>"

So với T, ta có:

→ T = "klm<sub>p</sub>opq" > "klm<sub>n</sub>opr"

Ngoài ra, các đáp án như "klmnop<sub>s</sub>", "klmnop<sub>t</sub>", "klmnop<sub>z</sub>", "klmoopq", ... đều được chấp nhận.

# Ví dụ 3

## Input:

S = “abcde”

T = “abcdf”

## Output:

“No such string”

## Lí do:

Không tồn tại chuỗi sao cho **lớn hơn S** và **nhỏ hơn T**.



# Hướng dẫn giải

# Thứ tự từ điển là gì?

A nhỏ hơn B (B khác rỗng) xét theo thứ tự từ điển nếu thoả 1 trong 2 điều kiện:

- A là **tiền tố** của B (ví dụ: A = “bigo”, B = “bigocoding”).
  - Tồn tại p, x, y ( $x < y$ ) sao cho:
    - A = px.....
    - B = py.....
    - (Ví dụ A = “xy**a**c”, B = “xy**b**z”)
- Do hai xâu S, T và xâu đáp án phải có độ dài bằng nhau nên chúng ta chỉ quan tâm đến điều kiện 2.

# Ý tưởng:

Ta biến đổi xâu  $S$  thành xâu kế tiếp đứng liền sau  $S$  và có thứ tự từ điển lớn hơn  $S$ .  
(Tương tự ý tưởng tìm số liền sau.)

Ví dụ (số):  $S = 101 \rightarrow S = 102$

Ví dụ (xâu):  $S = \text{"aab"} \rightarrow S = \text{"aac"}$

➔ Dễ dàng thực hiện bằng cách xét mã ASCII

# Ý tưởng liệu có luôn đúng?

Ví dụ (số):  $S = 109 \rightarrow S = 110$

Ví dụ (xâu):  $S = \text{"aaz"} \rightarrow S = \text{???}$

➔ Tương tự như chữ số, ta cũng thực hiện “nhớ 1” cho “hàng bên trái”:

$S = \text{"aaz"} + 1$

$S = \text{"__a"} \rightarrow \text{nhớ 1}$

$S = \text{"aba"}$

# Thuật toán giải:

Chạy ngược chuỗi S từ kí tự cuối về kí tự đầu, xét hai trường hợp sau:

- Nếu gặp kí tự 'z' thì biến kí tự này thành ký tự 'a' và tiếp tục xét kí tự trước đó.
- Nếu gặp kí tự khác 'z' thì tăng kí tự này lên một bậc trong bảng chữ cái, nghĩa là từ 'a'  $\rightarrow$  'b', 'g'  $\rightarrow$  'h'. Ngay sau tăng lên một bậc thì dừng vòng lặp (dùng break).

Sau khi biến đổi xong hãy so sánh giữa chuỗi kết quả và T, nếu chuỗi kết quả khác chuỗi T (nghĩa là chuỗi nhỏ hơn T) thì in ra chuỗi kết quả, ngược lại in "No such string".

# Các bước giải:

Bước 1: đọc vào hai chuỗi S và T

Bước 2: tính chuỗi S + 1 theo thuật toán vừa nêu.

Bước 3: So sánh chuỗi kết quả với chuỗi T:

    Nếu chuỗi kết quả khác T thì xuất ra đáp án là chuỗi kết quả

    Ngược lại thì xuất ra đáp án là “No such string”

**Độ phức tạp:**  $O(|S| + |T|)$ , với  $|S|$  và  $|T|$  lần lượt là độ dài của hai chuỗi S và T

# Mã giả

# Mã giả

```
Read(S, T)
N = S.length()
for i in (N - 1) -> 0:
    if (S[i] == 'z'):
        S[i] = 'a'
    else:
        S[i] = S[i] + 1
        break
if (S == T):
    print("No such string")
else:
    print(S)
```



Thank you