

BÀI TẬP NÂNG CAO

Bài tập 1

Kiểm tra 2 danh sách khác nhau có thể sinh ra cùng một chuỗi hay không?

Input danh sách $A = (s_1, s_2, \dots, s_n)$, danh sách $B = (t_1, t_2, \dots, t_m)$

Output: Có, nếu tồn tại một chuỗi được sinh ra từ danh sách A và danh sách B

Ví dụ:

Input: $A = (110, 0011, 0110)$ và $B = (110110, 00, 110)$.

Output: có vì $s_2 s_3 s_1 = t_2 t_3 t_1$

Kiểm tra các trường hợp sau:

1. $A = (0011, 11, 1101)$, $B = (101, 011, 110)$

2. $A = (100, 0, 1)$, $B = (1, 100, 0)$

(Gợi ý: sử dụng *itertools* (`import itertools`))

Bổ sung kiến thức: Các cách nối 2 chuỗi trong Python

1. Sử dụng toán tử +:

```
str1 = "abc"
```

```
str2 = "def"
```

```
str3 = str1 + str2
```

```
print(str3) #kết quả: abcdef
```

```
number = 1
```

```
print(str3 + number) # thông báo lỗi must be str, not int
```

2. Sử dụng phương thức **join()** để nối các chuỗi trong danh sách (chuyển các phần tử trong ds thành chuỗi):

```
L = ['abc', 'def']
```

```
str1 = ''.join(L)
```

```
print(str1) # kết quả: abc def
```

Hướng dẫn thực hiện:

1. Tạo các hoán vị từ danh sách 1 (sử dụng `itertools.permutations`)

```
L1 = ['110', '0011', '0110']
```

```
L1_temp=list(itertools.permutations(L1, len(L1))) # tạo các hoán vị
```

```
S1 = set() # tạo tập hợp rỗng
```

```
for i in L1_temp:
```

```
    str1 = ''.join(list(i)) #mỗi hoán vị kết nối thành chuỗi
```

```
    S1.add(str1)
```

```
print(S1)
```

2. Tạo các hoán vị từ danh sách 2 (sử dụng `itertools.permutations`)

```
L2 = ['110110','00','110']
```

```
L2_temp=list(itertools.permutations(L2, len(L2)))
```

```
S2 = set()
```

```
for i in L2_temp:
```

```
    str2 = "".join(list(i))
```

```
    S2.add(str2)
```

```
print(S2)
```

3. Tìm giao của 2 tập hợp, nếu có phần tử trong tập giao từ 2 tập ban đầu thì in “YES”, ngoài ra in “NO”

```
S3=S1.intersection(S2) s
```

```
if (len(S3) != 0):
```

```
    print("YES")
```

```
else:
```

```
    print("NO")
```

Tìm hiểu thêm các phương thức trong `itertools`

Bài tập 2:

Chuỗi DNA được tạo thành từ sự kết hợp của các codon, là 3 ký tự khác nhau bất kỳ thuộc bộ ký hiệu {A,C,G,T}, ví dụ ACT, ACG, TCG... Một đoạn gen là tập hợp của ít nhất 3 codon bắt đầu là ATG và kết thúc bằng TAA, TAG hoặc TGA.

Hãy viết chương trình kiểm tra 1 đoạn gen có hợp lệ hay không?

Ví dụ:

Input: ATGCCCTAG

Output: không hợp lệ

Input: ATGCGTTGA

Output: hợp lệ