

BOJ 24525

SKK 문자열

소프트웨어학과 201921017 이지우

1. 접근

문제

포함된 K의 개수가 S의 개수의 정확히 2배이면서, S와 K가 적어도 한 번은 등장하는 문자열을 SKK 문자열이라고 한다.

SKK 문자열은 S, K 말고도 다른 알파벳 또한 포함할 수 있다.

알파벳 대문자로만 이루어진 문자열 S가 주어질 때, S의 부분 문자열 중 길이가 가장 긴 SKK 문자열을 찾는 프로그램을 작성하라.

입력

첫째 줄에 길이가 1이상 100,000 이하인 알파벳 대문자로만 이루어진 문자열 S가 주어진다.

출력

S의 부분 문자열 중 길이가 가장 긴 SKK 문자열의 길이를 출력한다. 만약 그러한 문자열이 없으면 -1을 출력한다.

1. 접근

알고리즘은

* DP

* Prefix sum

구간

누적합

~~S = -2~~
~~K = 1~~

Prefix sum : 구간 합을 구할 때 사용하는 알고리즘

예를 들어 배열 arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]일 때

arr[a:b]의 합을 빠르게 구하기 위해 구간 합을 미리 저장해두는 방식!

pre-sum 1 2
1 1 1

prefix_s[i] : [0, i] 구간에 존재하는 문자 S의 개수

prefix_k[i] : [0, i] 구간에 존재하는 문자 K의 개수

S & K
1 1

prefix_s[b] - prefix_s[a-1] : [a, b] 구간에 존재하는 문자 S의 개수

prefix_k[b] - prefix_k[a-1] : [a, b] 구간에 존재하는 문자 K의 개수

a ~ b - a-1 a ~ b

DP[i] : i 위치까지 왔을 때 최장 SKK 문자열의 길이

1. 접근

구간을 나타낼 수 있어야 하기 때문에
이중 for문을 써보자! (응안돼~)

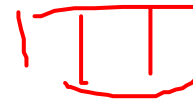
$$\begin{aligned} \text{num_s} &= \text{prefix_s}[b] - \text{prefix_s}[a-1] \\ \text{num_k} &= \text{prefix_k}[b] - \text{prefix_k}[a-1] \end{aligned}$$

$\text{num_s} * 2 = \text{Num_k}$ 이면 $5k$
 $\text{DP}[b] = \max(\text{DP}[b], b-a+1)$ # $b-a+1$: $[a, b]$ 구간의 길이
여하튼 함 해보자

2. 구현 - 입력 ^^

```
s = input()

if len(s) < 3:
    print(-1)
```



2. 구현

```
else:

    prefix_s = [0] * len(s)
    prefix_k = [0] * len(s)

    for i in range(len(s)):
        if i != 0:
            prefix_s[i] = prefix_s[i-1]
            prefix_k[i] = prefix_k[i-1]

        if s[i] == 'S':
            prefix_s[i] += 1
        elif s[i] == 'K':
            prefix_k[i] += 1
```

```
dp = [0] * len(s)
```

```
for a in range(len(s)):
```

```
    for b in range(a, len(s)):
```

```
        if a == 0:
```

```
            num_s = prefix_s[b]
```

```
            num_k = prefix_k[b]
```

```
        else:
```

```
            num_s = prefix_s[b] - prefix_s[a-1]
```

```
            num_k = prefix_k[b] - prefix_k[a-1]
```

```
        if num_s == 0 or num_k == 0:
```

```
            continue
```

```
        if num_s * 2 == num_k:
```

```
            dp[b] = max(dp[b], b-a+1)
```

```
        else: dp[b] = dp[b-1]
```

```
print(dp[-1] if dp[-1] else -1)
```

3. 결과

아이디	문제	문제 제목	결과	메모리	시간	언어	코드 길이
january	24525	SKK 문자열	시간 초과			PyPy3	884 B

어렵다..

사다리타기나 하자!

(͡° ͜ʖ ͡°) ͡⌋

소프트웨어학과 201921017 이지우