

# Стринг в стринг

Намерете колко пъти стрингът S1 се повтаря в стринга S2. Стринговете са много дълги и началото на S1 се повтаря на много места в S2.

Жокер: Това е класическата задача за приложение на техниката хеширане и алгоритъма Rolling hash за сравнение на стрингове. Ето и няколко ресурса за Rolling hash алгоритъма:

<https://www.geeksforgeeks.org/string-hashing-using-polynomial-rolling-hash-function/> <https://iq.opengenus.org/rolling-hash/> [https://en.wikipedia.org/wiki/Rolling\\_hash](https://en.wikipedia.org/wiki/Rolling_hash)

Първо пресметнете хеша на S1 използвайки рекурентната формула:  $H[i] = H[i-1] * \text{SOME\_BIG\_PRIME\_NUMBER} + S1[i]$  при начално условие  $H[-1] = 0$  или  $H_{\text{next}} = H_{\text{previous}} * \text{SOME\_BIG\_PRIME\_NUMBER} + S1[i]$  при начално условие  $H_{\text{previous}} = 0$ . След това приложете алгоритъма Rolling hash върху S2: За индекс по-малък от  $S1.size()$  смятайте нормален хеш (започващ от първият му символ) на S2 по стандартната формула  $H2[i] = H2[i-1] * \text{SOME\_BIG\_PRIME\_NUMBER} + S2[i]$ , а от там до края смятайте Rolling hash премахвайки значението на най-левия символ, който вече е извън прозореца от последните  $S1.size()$  на брой символа:  $H2[i] = H2[i-1] * \text{SOME\_BIG\_PRIME\_NUMBER} + S2[i] - S2[i - S1.size()] * \text{pow}(\text{SOME\_BIG\_PRIME\_NUMBER}, S1.size())$ . Преизчислете  $\text{pow}(\text{SOME\_BIG\_PRIME\_NUMBER}, S1.size())$  в самото начало на програмата (веднага след като получите S1 от конзолата). В други по-трудни задачи с Rolling hash  $\text{hashpow}(\text{SOME\_BIG\_PRIME\_NUMBER}, i)$  трябва да се преизчисли за всички възможни  $i=0..\text{MAX\_STRING\_LEN\_IN\_THE\_CURRENT\_PROBLEM}$ . В нашата задача  $\text{MAX\_STRING\_LEN\_IN\_THE\_CURRENT\_PROBLEM}$  би било 1,000,000.  $\text{SOME\_BIG\_PRIME\_NUMBER}$  трябва да е еднакъв при пресмятането на хешовете за S1 и S2. Добра стойност за  $\text{SOME\_BIG\_PRIME\_NUMBER}$  е 1,000,000,007, но може и да е и друго число, което да не е просто. Докато пресмятате  $H2[i]$  го сравнявате с вече пресметнатия  $H[S1.size() - 1]$  и при съвпадение сте открили (с много голяма вероятност - 0.9999...) подстринг от S2 (с дължина  $S1.size()$ , започващ от символ  $S2[i - S1.size() + 1]$  и завършващ на символ  $S2[i]$ ), който е равен на S1.

Забележка: в някои статии  $H2[i] = H2[i-1] * \text{SOME\_BIG\_PRIME\_NUMBER} + S2[i] - S2[i - S1.size()] * \text{pow}(\text{SOME\_BIG\_PRIME\_NUMBER}, S1.size())$  е записано

$H2[i] = (H2[i-1] - S2[i - S1.size()] * \text{pow}(\text{SOME\_BIG\_PRIME\_NUMBER}, S1.size()) -$

1))\*SOME\_BIG\_PRIME\_NUMBER + S2[i]

Сами виждате, че единия израз е получен след разкриването на скобите на другия

### **Input Format**

На първият ред ще получите стринга S1, а на втория S2

### **Constraints**

S1.length() <= 1,000,000 S2.length() <= 1,000,000

### **Output Format**

Отпечатайте броя на повторенията на S1 в S2.

### **Sample Input 0**

```
abc  
quweabcdfqwdjabcghevqwabc
```

### **Sample Output 0**

```
3
```