

Rabin-Karp algorithm in Java → The number of equal substring pairs

5 users solved this problem. Latest completion was 22 days ago.

Hard

43 minutes

Share:

You've skipped this problem 1 day ago.
Maybe now is a good time to try it out again?

Code Challenge — Write a program

Given a string s and a set of pairs, where each element represents the starting and the ending position of a substring of s . Write a program that counts the number of pairs of equal substrings.

Input: the first line contains a string s . The second line contains an integer t . Each of the following t lines contains 4 integers i, j, k and m separated by space, such that $0 \leq i < j \leq |s|, 0 \leq k < m \leq |s|$.

Output: the number of pairs, such that $s[i : j]$ is equal to $s[k : m]$.

In this problem for a polynomial hash with constants $a = 53$ and $m = 10^9 + 9$ it is guaranteed that if hash values for two substrings are equal, then the strings are equal as well.

Hint 1: calculate hash values of all prefixes of s and then use them to find a hash value for any substring of s .

Hint 2: for a substring $s[i : j]$ a polynomial hash is calculated as follows:

$$hash(s[i : j]) = \left(\sum_{k=i}^j s[k] \cdot a^{k-i} \right) \bmod m.$$

Multiplying the equality by a^i we will get the following:

$$hash(s[i : j]) \cdot a^i = \left(\sum_{k=i}^j s[k] \cdot a^k \right) \bmod m = \left(\left(\sum_{k=0}^j s[k] \cdot a^k \right) - \left(\sum_{k=0}^{i-1} s[k] \cdot a^k \right) \right) \bmod m.$$

That is,

$$hash(s[i : j]) \cdot a^i = (hash(s[0 : j]) - hash(s[0 : i - 1])) \bmod m.$$

So, if we know hash values for all prefixes of s , we can calculate a hash value for any substring of s . The only problem is that we get a hash value multiplied by a^i . Then, if we want to compare hash values of $s[i : i']$ and $s[j : j']$ (assuming $i \leq j$) we need to multiply the first one by a^{j-i} .

Sample Input 1:

```
abacabad
2
1 4 5 8
0 3 4 7
```

Sample Output 1:

```
1
```

Sample Input 2:

```
seamlessness
3
4 8 8 12
5 8 9 12
0 1 11 12
```

Sample Output 2:

```
2
```

Code Editor

IDE

Java

```
1 public class Main {
2     // your code here
3 }
```

Code snippets from theory

Run

Start again (reset)

 Code snippets from theory

Skip problem

Time limit: 5 seconds Memory limit: 256 MB

Comments (10)

Hints (0)

Useful links (0)

Solutions (0)


Share something, Sergey Kubatko

Sort by: Last posted

- SK

Sergey Kubatko 1 day ago

spent two days, algo is right, cannot pass test#6 as Scanner reads only 986895 chars instead of 2M, pls give a clue how it can be solved

 0

[Reply](#)
- AN

Adrian Nachev 4 months ago [Report](#)


Would be nice to say in the downloaded tests what is the expected result and what was the actual result. Otherwise a test with a massive string is completely pointless.

 0

[Reply](#)
- AS

Andrey Shinkaryov 4 months ago [Report](#)

16mb string) Got correct answer on computer, but runtime error here.

 0


[Reply](#)
- J

JoyceS 5 months ago [Report](#)

1. we need clarification on including/excluding the indexes
in all equations of the hints, substring `s[i..j]` means from index `i` to index `j` inclusive. however, in the example input, the right index is given as exclusive.

2. the modulo operations are more complicated than simple multiplying
ref: https://en.wikipedia.org/wiki/Modulo_operation#Equivalences
 $(a \cdot b) \bmod m = ((a \bmod m) \cdot (b \bmod m)) \bmod m$

3. test case #6 is a HUGE one, maybe we need to figure out a smarter way for all the calculations and storage?
would be much more helpful if we can have an expected output

 0


[Reply](#)
- LA

LAURENT APICELLA 5 months ago [Report](#)

In all the equalities with a sigma sum sign, it is `j-1` instead of `j`, and in the last equality:

 $\text{hash}(s[i..j]) \cdot a_i = (\text{hash}(s[0..j]) - \text{hash}(s[0..i-1])) \bmod m$


it is `i` instead of `i-1`

 0

[Reply](#)
- A

andioz 5 months ago [Report](#)

Ok, I think I am struggling with this: you corrected the definition with Timurs comment, now `j` means exclusive index, right?
But for me the formulas still include `j!` Am i correct?

 0

[Reply](#)
- A

andioz 5 months ago [Report](#)

I don't understand the last formula, maybe a bug? " $\text{hash}(s[i..j]) \cdot a_i = (\text{hash}(s[0..j]) - \text{hash}(s[0..i-1])) \bmod m$ " - but "`i`" can start at 0, so "`i - 1`" -> -1 sound wrong to me?

 0

[Reply](#)
- A

andioz 5 months ago [Fixed](#)

k has two meanings in explanation. use different letters.

♡ 0

TI **Timur Ishakov** 5 months ago Fixed

* do not

♡ 0

TI **Timur Ishakov** 5 months ago Fixed

Samples do match input definition in task. Task states that $0 \leq i < j < |s|, 0 \leq k < m < |s|$, but in sample1 - $|s| = 8$, and m is 8 in first

♡ 0 [Show all](#)