Problem of the day. Next problem in **14h 14m 9s**

You will get + 💎 **5–10** for solving it.

Java → Implementation of basic algorithms → Rabin-Karp algorithm in Java

Rabin-Karp algorithm in Java → Calculating a substring hash

Hard 1 hour

1 users solved this problem. Latest completion was **2 days ago**.

Code Challenge — Write a program

Before start solving this problem, let's learn a bit more theory about the polynomial hash function.

One property of the polynomial hash function

The polynomial hash function has the following property: if we know hash values for all prefixes of some string, we can find a hash value for any substring of this string in $O(1)$. In other words, if we precalculate hashes for prefixes of some string only once, we will get a fast way to compute a hash for arbitrary substrings of this string. Let's now learn the details of how the approach works.

Consider a string $s = s_0 s_1 ... s_{n-1}$. A polynomial hash for a substring $s[i:j]$ can be calculated as follows:

$$h_P(s[i:j]) = \left(s_i \cdot a^0 \ + s_{i+1} \cdot a^1 + ... + s_{j-1} \cdot a^{j-i-1}\right) \ mod \ m.$$

If we multiply the expression by a^i, we will get the following:

$$h_P(s[i:j]) \cdot a^i \ = \left(s_i \cdot a^i \ + s_{i+1} \cdot a^{i+1} \ + \ ... + s_{j-1} \cdot a^{j-1}\right) \ mod \ m = \left(\sum_{k=i}^{j-1} s_k \cdot a^k\right) \ mod \ m.$$

The last sum can be rewritten as a difference of two sums, each being a polynomial hash for a prefix of s:

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

Using the slice notation, the expression above can be written as follows:

$$h_P(s[i:j]) \cdot a^i = (h_P(s[0:j]) - h_P(s[0:i])) \ mod \ m.$$

This formula gives us an explicit way to calculate a hash value for any substring of s in $O(1)$ given that hashes for all prefixes are precalculated.

Formulating the problem

In this task, you need to apply the described approach to calculate hash values for given substrings of some string. One thing to keep in mind here is that we got hash value multiplied by a^i. Thus, to get the hash value itself, we need to perform a modulo division by a^i. For the sake of simplicity, we will avoid this step here and ask you to calculate hash values with this additional multiplier.

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

Output: the first line should contain hash values for all prefixes of s. The second line should contain k integers, each equal to $h_P(s[i:j]) \cdot a^i$ for the given in the input indexes i and j. Note that all hash values have to be non-negative.

In this problem, you are expected to use a polynomial hash with the following parameters: $a = 53$ and $m = 10^9 + 9$.

Sample Input 1:

```
abacabad
3
1 3
3 7
7 8
```

Sample Output 1:

```
33 1835 94532 5305227 265691100 484337736 908248414 508901936
94499 908153882 600653531
```

Code Editor IDE

✓ IDE is responding IntelliJ IDEA 2019.3.3

✓ EduTools plugin is responding 3.5-2019.3-4422

Solve in IDE

✓ **Correct, but can be improved**

Thanks for your feedback!

Write here how we could improve this problem

Continue View solutions (0)

Time limit: 8 seconds Memory limit: 256 MB

This content was created 21 days ago and updated about 4 hours ago. Share your feedback below in comments to help us improve it!

Show discussion (0)