**String Theory**

**Sliding window Algorithm**

Checks whether the pattern exist in the string .

Look through the window of length of pattern and compare each letter.

**String Hashing**

Checks 2 strings are equal or not in O(1)

hash(s)=s[0]+s[1]⋅p+s[2]⋅p^2+...+s[n−1]⋅p^n−1modm

=∑i=0n−1s[i]⋅p^I modm,

P is a number greater then alphabets generally 31for English alphabets and 53 for lower and upper case .

M is larege number in 10 to the power 9.

**Substring Hashing in order 1 time**

Consider string Harshith .

After hashing it will generate its hash value .

Consider substring “arsh” now to calculate hash value of arsh we can do is

Hash(“arsh”) = p^0(a) + p^1(r) + p^2(s) + p^3(h);

We can do is

Find the contribution in main string and divide it by the p to the power starting index of substring.

Now comes the modulo part

ie we have to take module ,but division under module is not possible

So well find the multiplicative inverse of the number and take the mod.

Refer number theory concepts.

Rabin-karp Algorithm

String =”coding”;

Substring = “din“

Hash(string) =x;

Hash(substring) = (contribution in the main string )/p to the power the index of l;

Ie

Hash(“din”) = d.p^0 + i.p^1 + n.p^2 = (d.p^2 + i.p^3 + n.p^3)/p^2

(d.p^0 + i.p^1 + n.p^2)\*p^2 = (d.p^2 + i.p^3 + n.p^3)