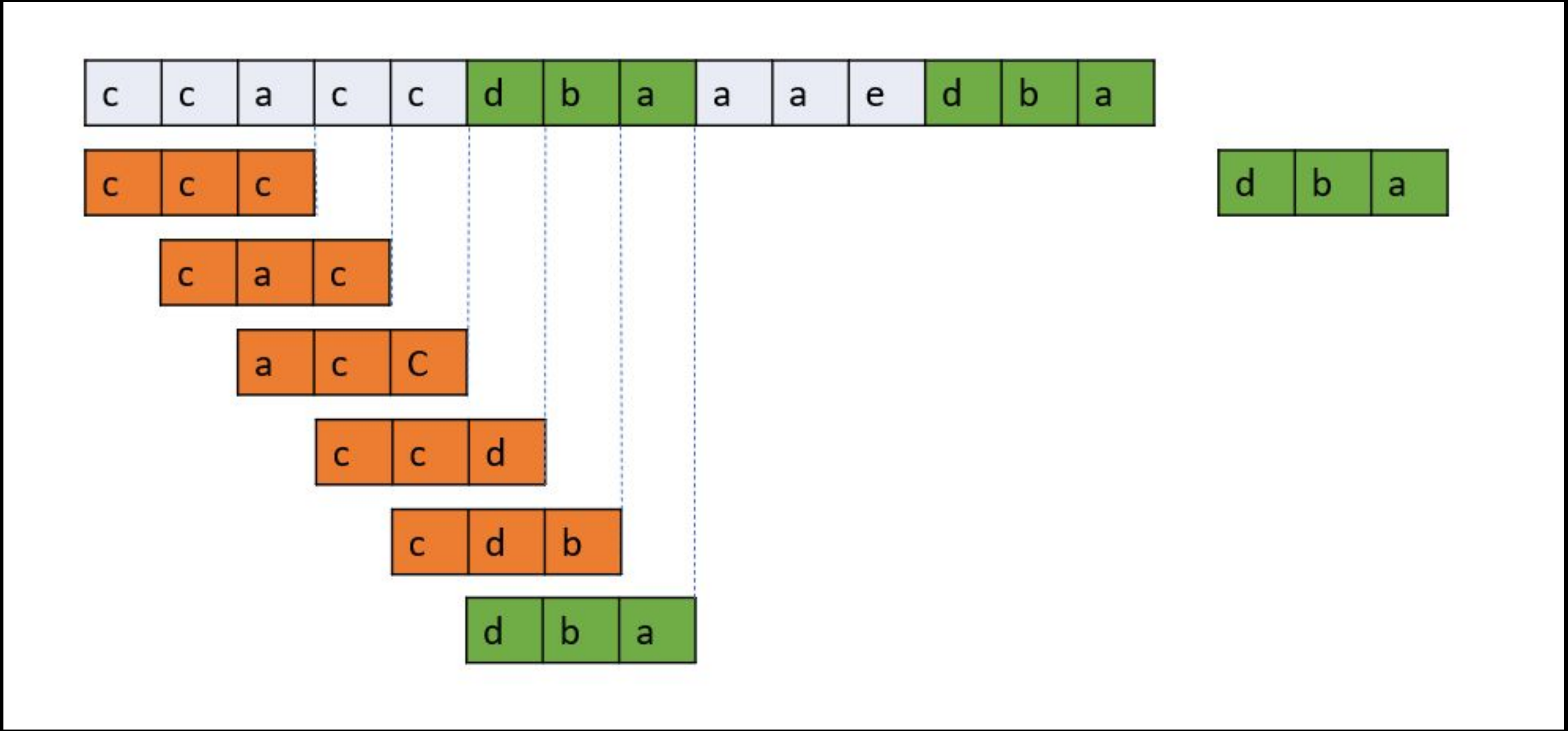
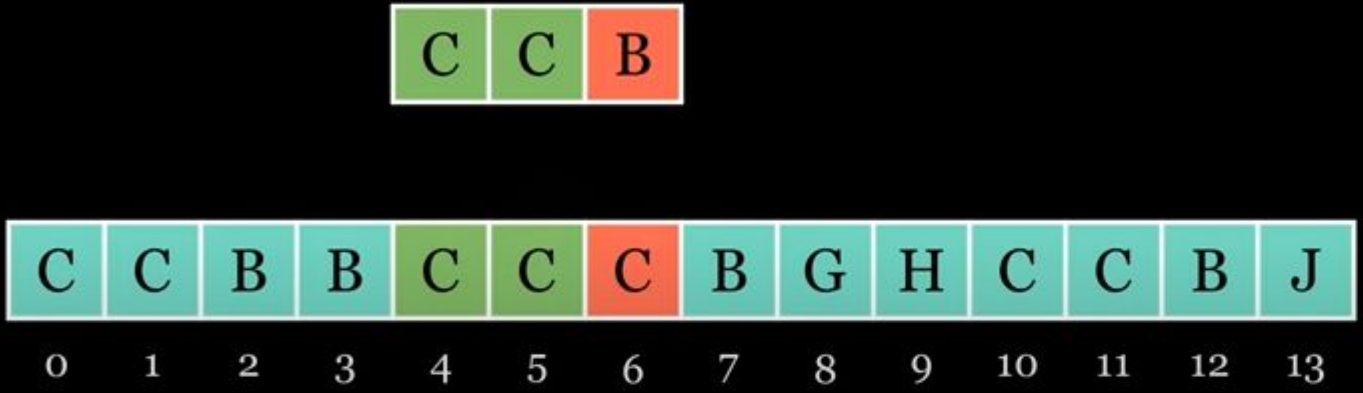
Rabin Karp

Rodrigo Li

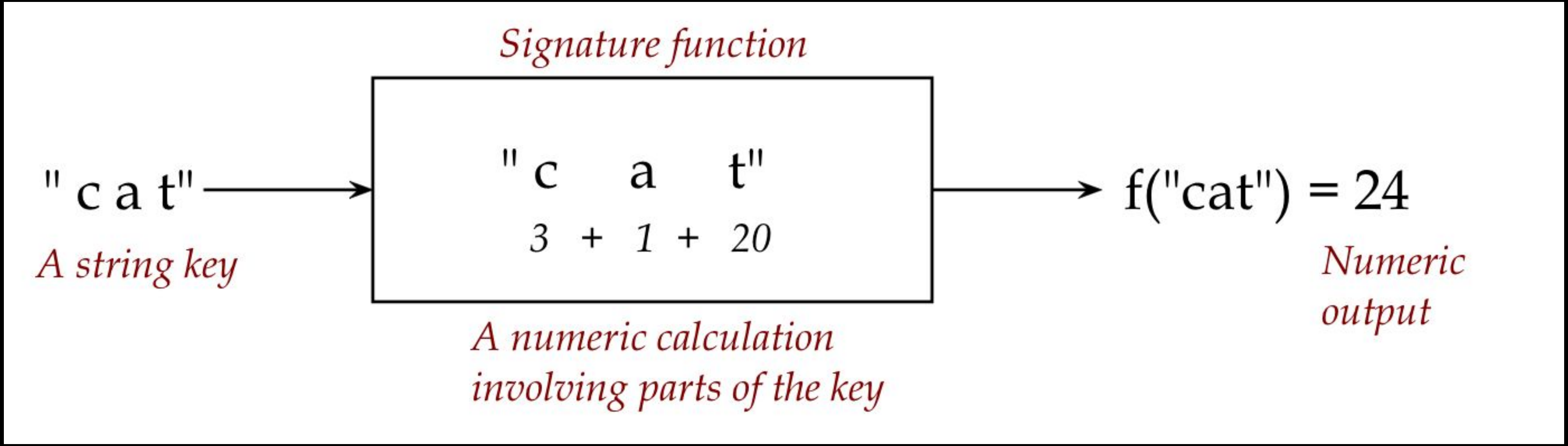
El problema



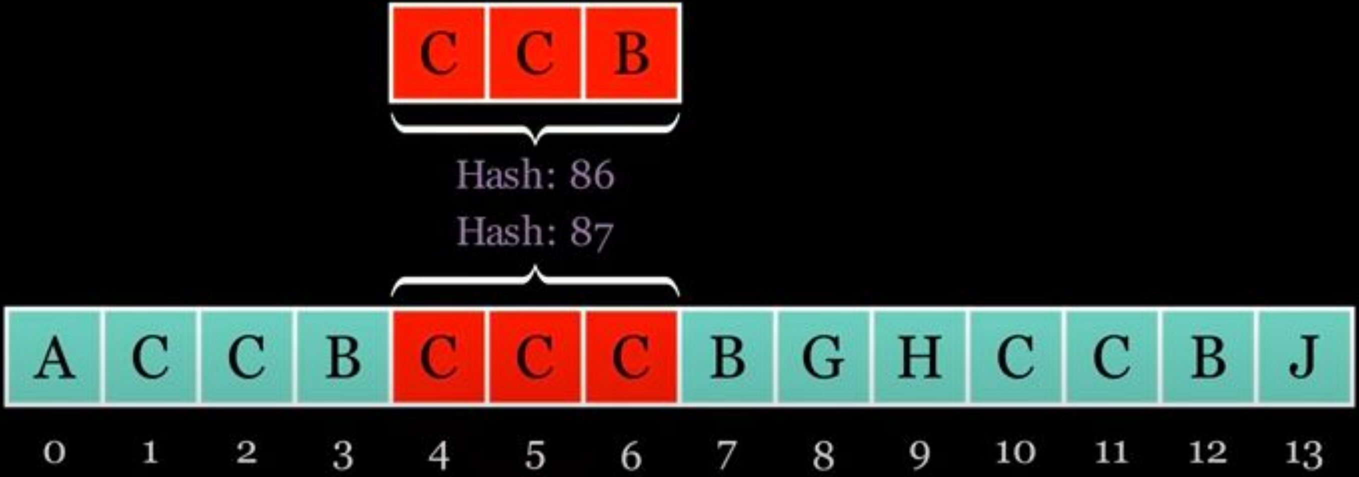
Primera intuición



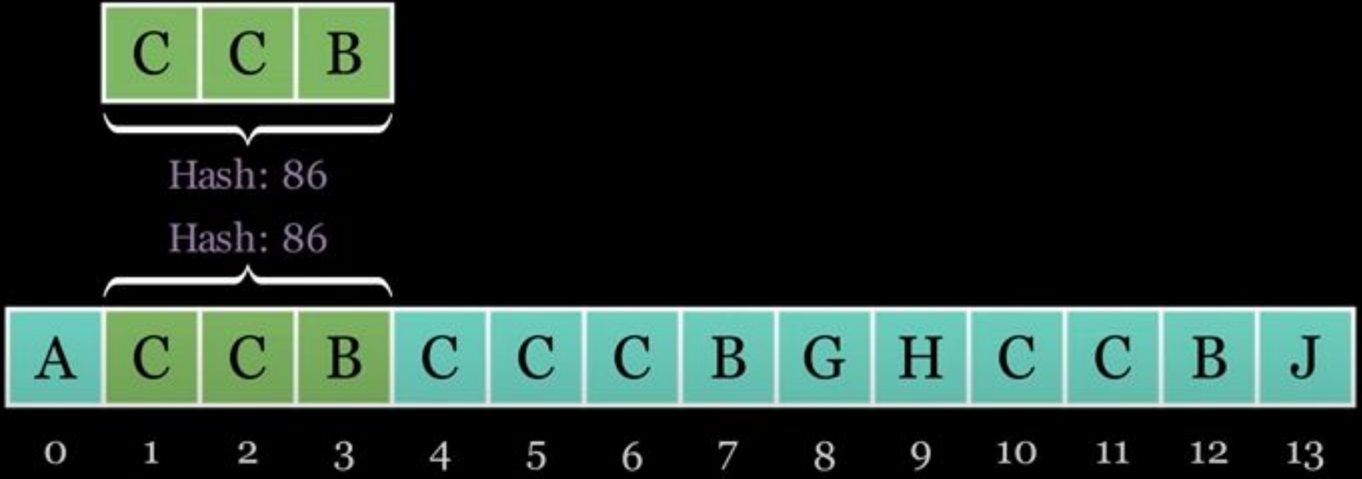
IDEA: función hash



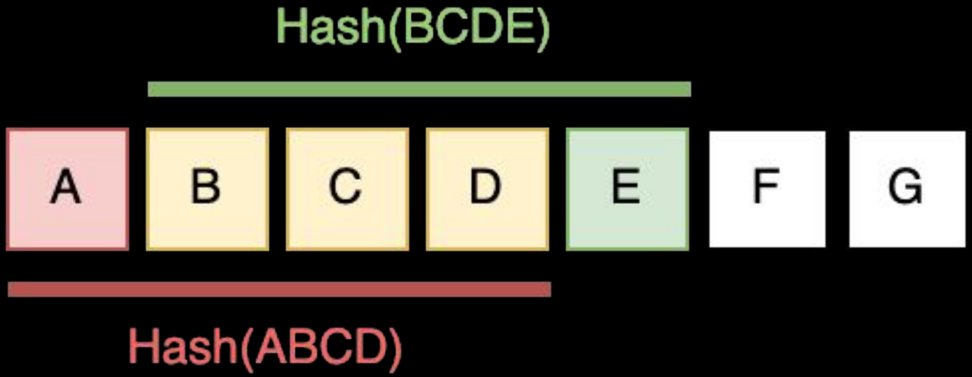
Si los hashes no coinciden



Si los hashes coinciden



Rolling Hash

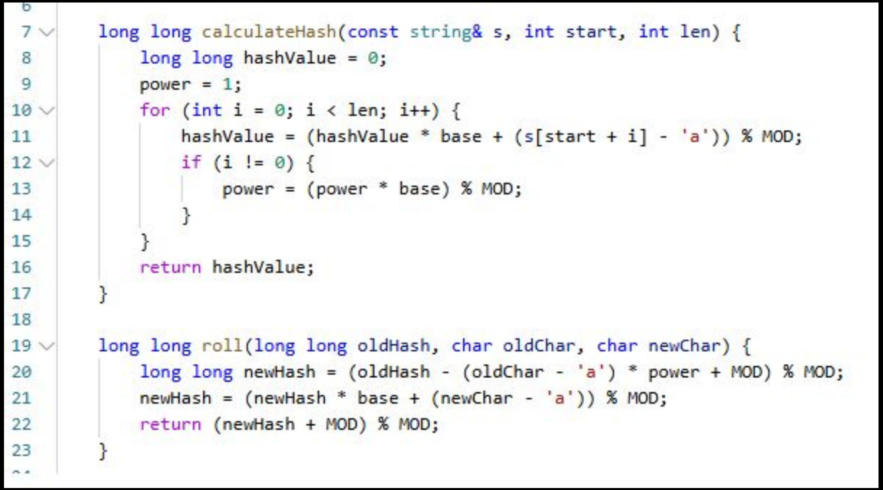


Función de Rolling Hash

$$H = c_1 \times b^{n-1} + c_2 \times b^{n-2} + \cdots + c_n \times b^0$$

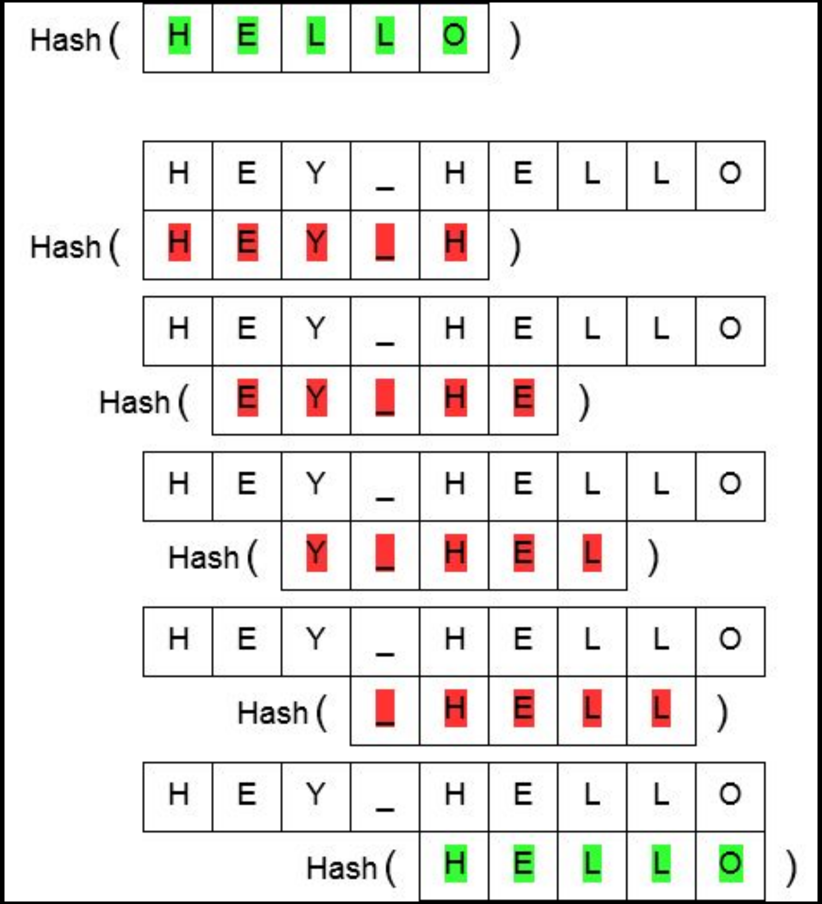
$$H_{\text{new}} = (b \times (H_{\text{old}} - c_{\text{old}} \times b^{n-1}) + c_{\text{new}})$$

Finalmente:



```
long long calculateHash(const string& s, int start, int len) {
    long long hashValue = 0;
    power = 1;
    for (int i = 0; i < len; i++) {
        hashValue = (hashValue * base + (s[start + i] - 'a')) % MOD;
        if (i != 0) {
            power = (power * base) % MOD;
        }
    }
    return hashValue;
}

long long roll(long long oldHash, char oldChar, char newChar) {
    long long newHash = (oldHash - (oldChar - 'a') * power + MOD) % MOD;
    newHash = (newHash * base + (newChar - 'a')) % MOD;
    return (newHash + MOD) % MOD;
}
```


Hash (H E L L O)
H E Y _ H E L L O
Hash (H E Y _ H)
H E Y _ H E L L O
Hash (E Y _ H E)
H E Y _ H E L L O
Hash (Y _ H E L)
H E Y _ H E L L O
Hash (_ H E L L)
H E Y _ H E L L O
Hash (H E L L O)