String Searching Algorithms

Αδάμος Ττοφαρή

Περιεχόμενα

- The Problem
- Naïve Approach
- KMP Algorithm
  - Visualization
  - Implementation
- Hashing
  - Hash Functions
  - Modulus
- Rabin-Karp Algorithm
  - Visualization
  - Implementation
- Rabin-Karp vs KMP

The Problem

Το βασικό String Searching (matching) ορίζεται ως εξής: Έστω έχουμε 2 string - το κείμενο (Text) και το πρότυπο (Pattern) και πρέπει να καθορίσουμε αν το πρότυπο εμφανίζεται στο κείμενο. Το πρόβλημα είναι γνωστό και ως “the needle in a haystack problem.”

Naïve Approach

Η απλή λύση (Brute Force) είναι να θεωρήσουμε κάθε χαρακτήρα στο κείμενο ως αρχή του pattern και ελέγχουμε χαρακτήρα-χαρακτήρα αν έχουμε ταίρι.

Πολυπλοκότητα: O(N*M)

```
for(int i=0; i<text.size()-pattern.size(); i++){
    bool match=true;
    for(int j=0; j<pattern.size(); j++){
        if(pattern[j]!=text[i+j]){
            match=false;
            break;
        }
    }
    if(match)
        return match;
}
return false;
```

Knuth-Morris-Pratt

Βασική αδυναμία του απλού τρόπου είναι ότι χρειάζεται να ξεκινούμε από την αρχή του pattern για κάθε χαρακτήρα του text. Ο KMP έχει το πλεονέκτημα ότι υπολογίζει άπο ποιο σημείο του pattern να ξεκηνήσει για να ελέγξει αν υπάρχει match. Με την βοήθεια ενός νέου String s=patternt+'#'+text και ένος πίνακα που οι τίμες του είναι το μέγιστο μήκος του Suffix σε κάθε index του πίνακα που είναι ίσο με το μέγιστο prefix του substring έως το index-1.

Visualization

Αρχικά έχουμε το καινούριο String s=pattern+'#'+text με τον παράλληλο του πίνακα με τα prefixes

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~																

Visualization

Θέτω το prefix του 1ου χαρακτήρα -1

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1															

Visualization

Βλέπουμε το prefix της προηγούμενης θέσης.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1															

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1															

Visualization

Λόγο του ότι είναι άνισοι και ότι το prefix του προηγουμένου βρίσκεται στο -1 τότε θέτουμε το prefix -1

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1														

Visualization

Βλέπουμε το prefix της προηγούμενης θέσης.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1														

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1														

Visualization

Αφού είναι ίσοι προσθέτουμε 1 στην τιμή του prefix

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0													

Visualization

Βλέπουμε το prefix της προηγούμενης θέσης.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0													

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0													

Visualization

Αφού είναι ίσοι προσθέτουμε 1 στην τιμή του prefix.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1												

Visualization

Βλέπουμε το prefix της προηγούμενης θέσης.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1												

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1												

Visualization

Λόγο του ότι είναι άνισοι αναδρομικά πηγαίνουμε στο prefix του προηγουμένου prefix

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1												

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1												

Visualization

Λόγο του ότι είναι άνισοι και ότι
το prefix του προηγουμένου βρίσκεται
στο -1 τότε θέτουμε το prefix -1

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1											

Visualization

Βλέπουμε το prefix της προηγούμενης θέσης.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1											

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1											

Visualization

Λόγο του ότι είναι άνισοι και ότι
το prefix του προηγουμένου βρίσκεται
στο -1 τότε θέτουμε το prefix -1

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1										

Visualization

Βλέπουμε το prefix της προηγούμενης θέσης.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1										

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1										

Visualization

Λόγο του ότι είναι άνισοι και ότι
το prefix του προηγουμένου βρίσκεται
στο -1 τότε θέτουμε το prefix -1

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1									

Visualization

Βλέπουμε το prefix της προηγούμενης θέσης.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1									

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1									

Visualization

Λόγο του ότι είναι άνισοι και ότι
το prefix του προηγουμένου βρίσκεται
στο -1 τότε θέτουμε το prefix -1

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1								

Visualization

Βλέπουμε το prefix της προηγούμενης θέσης.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1								

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1								

Visualization

Αφού είναι ίσοι προσθέτουμε 1 στην τιμή του prefix.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0							

Visualization

Βλέπουμε το prefix της προηγούμενης θέσης.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0							

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0							

Visualization

Αφού είναι ίσοι προσθέτουμε 1 στην τιμή του prefix.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1						

Visualization

Βλέπουμε το prefix της προηγούμενης θέσης.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1						

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1						

Visualization

Αφού είναι ίσοι προσθέτουμε 1 στην τιμή του prefix.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2					

Visualization

Βλέπουμε το prefix της προηγούμενης θέσης.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2					

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2					

Visualization

Αφού είναι ίσοι προσθέτουμε 1 στην τιμή του prefix.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3				

Visualization

Βλέπουμε το prefix της προηγούμενης θέσης.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3				

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3				

Visualization

Λόγο του ότι είναι άνισοι αναδρομικά πηγαίνουμε στο prefix του προηγουμένου prefix

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3				

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3				

Visualization

Αφού είναι ίσοι προσθέτουμε 1 στην τιμή του prefix.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3	2			

Visualization

Βλέπουμε το prefix της προηγούμενης θέσης.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3	2			

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3	2			

Visualization

Αφού είναι ίσοι προσθέτουμε 1 στην τιμή του prefix.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3	2	3		

Visualization

Βλέπουμε το prefix της προηγούμενης θέσης.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3	2	3		

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3	2	3		

Visualization

Λόγο του ότι είναι άνισοι αναδρομικά πηγαίνουμε στο prefix του προηγουμένου prefix

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3	2	3		

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3	2	3		

Visualization

Λόγο του ότι είναι άνισοι αναδρομικά πηγαίνουμε στο prefix του προηγουμένου prefix.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3	2	3		

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3	2	3		

Visualization

Λόγο του ότι είναι άνισοι και ότι
το prefix του προηγουμένου βρίσκεται
στο -1 τότε θέτουμε το prefix -1

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3	2	3	-1	

Visualization

Βλέπουμε το prefix της προηγούμενης θέσης.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3	2	3	-1	

Visualization

Ελέγχουμε αν ο επόμενος χαρακτήρας από την θέση του προηγουμένου prefix είναι ίσος με τον παρόν χαρακτήρα.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3	2	3	-1	

Visualization

Λόγο του ότι είναι άνισοι και ότι
το prefix του προηγουμένου βρίσκεται
στο -1 τότε θέτουμε το prefix -1.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3	2	3	-1	-1

Visualization

Μετά από μια
διάσχιση έχουμε υπολογίσει τις τιμές
που αντιστοιχούν στα prefixes.

-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
~	n	a	n	a	#	i	v	a	n	a	n	a	n	a	z	a
~	-1	-1	0	1	-1	-1	-1	-1	0	1	2	3	2	3	-1	-1

Implementation

```
void kmp(string text, string pattern){
    string s=pattern+"#"+text;
    int pre[s.size()];
    pre[0]=-1;
    for(int i=1; i<s.size(); i++){
        int p1=pre[i-1], p2=i-1;
        while(p2!=-1 && s[p1+1]!=s[i]){
            p2=p1;
            p1=pre[max(p1, 0)];
        }
        if(p2==-1)
            pre[i]=-1;
        else
            pre[i]=pre[p2]+1;
    }
}
```

Hashing

Hashing είναι όταν δεχόμαστε δεδομένα (string σε αυτή την περίπτωση) αγνώστου μεγέθους και με την βοήθεια μιας hash function μετατρέπεται σε ακέραιο αριθμό. Με τέτοιο τρόπο ξεφορτωνόμαστε τον O(N) έλεγχο μεταξύ των string.

Hash Function

Το hash function που θα χρησημοποιησουμε είναι της μορφης.

$$s_0 \cdot q^{n-1} + s_1 \cdot q^{n-2} + s_2 \cdot q^{n-3} + \ldots + s_{n-2} \cdot q^1 + s_{n-1}$$

Οπου το **S_i** αντιστοιχεί στον ανάλογο χαρακτήρα του string ('a'=1, 'b'=2 … 'z'=26) **q** αντιστοιχεί στην βάση του Hash Function.

Modulus

Λόγο του ότι ο υπολογιστής υπολογίζει μέχρι 64-bit αριθμούς είναι αδύνατο να υπολογίζει hash functions μεγάλων string για αυτό χρησιμοποιούμε έναν μεγάλο πρώτο αριθμό για να υπολογίζουμε το υπόλοιπο και να ελπίζουμε να μην υπάρξουν Colisions ή να αφήσουμε την μηχανή να κάνει το υπόλοιπο με το **18,446,744,073,709,551,616.**

Rabin-Karp

Μια άλλη αδυναμία του βασικού αλγόριθμου είναι ότι για κάθε φόρα που μετατοπίζουμε το pattern πρέπει να ξανά περνούμε από κάθε χαρακτήρα. Αλλά με την βοήθεια του hashing και του sliding window, μπορούμε να ελέγχουμε άμεσος ισούνται το substring και το pattern.

Visualization

q=31

Pattern="nana"

Text="ivanananaza"

Έστω χρησιμοποιούμε ως δεδομένα το pattern="nana" και text="ivanananaza"

Visualization

q=31

Pattern="nana"

Text="ivanananaza"

Υπολογίζουμε το Hash Value του Pattern και των πρώτων K χαρακτήρων όπου K το μήκος του Pattern

Hash("nana")=418470

Hash("ivan")=289306

Visualization

Ελέγχουμε αν οι τιμές ισούνται

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("ivan")=289306

Visualization

Φεύγουμε τον 1ο χαρακτήρα από
το κομμάτι που ελέγχουμε

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("van")=Hash("ivan")-'i'*q^(k-1)=289306-9*31^3=21187

Visualization

Υπολογίζουμε το επόμενο κομμάτι πολλαπλασιάζοντας το μικρό κομμάτι με q (shift) και προσθέτουμε τον επόμενο χαρακτήρα

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("vana")=Hash("van")*q+'a'=21187*31+1=656798

Visualization

Ελέγχουμε αν οι τιμές ισούνται

q=31

Pattern="nana"

Text="ivanananaza"

i	v	a	n	a	n	a	n	a	z	a

Hash("nana")=418470

Hash("vana")=656798

Visualization

Φεύγουμε τον 1ο χαρακτήρα από το κομμάτι που ελέγχουμε

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("ana")=Hash("vana")-'v'*q^(k-1)=656798-22*31^3=1396

Visualization

Υπολογίζουμε το επόμενο κομμάτι πολλαπλασιάζοντας το μικρό κομμάτι με q (shift) και προσθέτουμε τον επόμενο χαρακτήρα

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("anan")=Hash("ana")*q+'n'=1396*31+14=43290

Visualization

Ελέγχουμε αν οι τιμές ισούνται

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("anan")=43290

Visualization

Φεύγουμε τον 1ο χαρακτήρα από το κομμάτι που ελέγχουμε

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("nan")=Hash("anan")-'a'*q^(k-1)=43290-1*31^3=13499

Visualization

Υπολογίζουμε το επόμενο κομμάτι πολλαπλασιάζοντας το μικρό κομμάτι με q (shift) και προσθέτουμε τον επόμενο χαρακτήρα

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("nana")=Hash("nan")*q+'a'=13499*31+1=418470

Visualization

Ελέγχουμε αν οι τιμές ισούνται

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("nana")=418470

Visualization

Φεύγουμε τον 1ο χαρακτήρα από το κομμάτι που ελέγχουμε

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("ana")=Hash("nana")-'n'*q^(k-1)=418470-14*31^3=1396

Visualization

q=31

Pattern="nana"

Text="ivanananaza"

Υπολογίζουμε το επόμενο κομμάτι πολλαπλασιάζοντας το μικρό κομμάτι με q (shift) και προσθέτουμε τον επόμενο χαρακτήρα

Hash("nana")=418470

Hash("anan")=Hash("ana")*q+'n'=1396*31+14=43290

Visualization

Ελέγχουμε αν οι τιμές ισούνται

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("anan")=43290

Visualization

Φεύγουμε τον 1ο χαρακτήρα από το κομμάτι που ελέγχουμε

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("nan")=Hash("anan")-'a'*q^(k-1)=43290-1*31^3=13499

Visualization

Υπολογίζουμε το επόμενο κομμάτι πολλαπλασιάζοντας το μικρό κομμάτι με q (shift) και προσθέτουμε τον επόμενο χαρακτήρα

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("nana")=Hash("nan")*q+'a'=13499*31+1=418470

Visualization

Ελέγχουμε αν οι τιμές ισούνται

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("nana")=418470

Visualization

Φεύγουμε τον 1ο χαρακτήρα από το κομμάτι που ελέγχουμε

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("ana")=Hash("nana")-'n'*q^(k-1)=418470-14*31^3=1396

Visualization

q=31

Pattern="nana"

Text="ivanananaza"

Υπολογίζουμε το επόμενο κομμάτι πολλαπλασιάζοντας το μικρό κομμάτι με q (shift) και προσθέτουμε τον επόμενο χαρακτήρα

Hash("nana")=418470

Hash("anaz")=Hash("ana")*q+'z'=1396*31+26=43302

Visualization

Ελέγχουμε αν οι τιμές ισούνται

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("anaz")=43302

Visualization

Φεύγουμε τον 1ο χαρακτήρα από το κομμάτι που ελέγχουμε

q=31

Pattern="nana"

Text="ivanananaza"

Hash("nana")=418470

Hash("naz")=Hash("anaz")-'a'*q^(k-1)=43302-1*31^3=13511

Visualization

Υπολογίζουμε το επόμενο κομμάτι πολλαπλασιάζοντας το μικρό κομμάτι με q (shift) και προσθέτουμε τον επόμενο χαρακτήρα

q=31

Pattern="nana"

Text="ivananaza"

Hash("nana")=418470

Hash("naza")=Hash("naz")*q+'z'=13511*31+1=418842

Visualization

Ελέγχουμε αν οι τιμές ισούνται

q=31

Pattern="nana"

Text="ivananaza"

Hash("nana")=418470

Hash("naza")=418842

Implementation

```
void RabinKarp(string pattern, string text){
    unsigned long long Q=1, q=31;
    unsigned long long pat=0, kom=0;
    int k=pattern.size();
    for(int i=0; i<k; i++){
        if(i!=0)
            Q*=q;
        pat=pat*q+(pattern[i]-'a'+1);
        kom=kom*q+(text[i]-'a'+1);
    }
    for(int i=k; i<text.length(); i++){
        kom-=(text[i-k]-'a'+1)*Q;
        kom*=q;
        kom+=text[i]-'a'+1;
        if(kom==pat)
            doSomething();
    }
}
```

KMP vs Rabin-Karp

KMP

- O(N+M)
- Δουλεύει πάντα
- Εκμεταλλεύεται τα prefixes

Rabin-Karp

- O(N+M)
- Υπάρχει μικρή πιθανότητα για Collisions
- Πιο ευέλικτος

“In theory, theory and practice are the same.
In practice, they’re not.”

Yoggi Berra

adamos2468@gmail.com