Rabin-Karp algorithm

Rabin-Karp-Matcher(T, P, d, q)

n = T.length
m = P.length
h = d^{m-1} \mod q
p = 0
t = 0

#preprocessing
for i = 1 to m
    p = (dp + P[i]) \mod q
    t = (dt + T[i]) \mod q

#matching
for s = 1 to n-m+1
    if p == t:
        if P[1...m] == T[s,...,s+m-1]
            print "Pattern occurs with shift" s
    if s < n-m+1:
        t = (d(t - T[s]h)+T[s+m])mod q
EXAMPLE

n : 21
m : 4
d : 10
Divisor, Q : 997
Pattern, P: 2284

p = 0
i = 1, p = (d × p + P[i]) mod Q = (10 × 0 + 2) mod 997 = 2 mod 997 = 2
i = 2, p = (10 × 2 + 2) mod 997 = 22
i = 3, p = (10 × 22 + 8) mod 997 = 228
i = 4, p = (10 × 228 + 4) mod 997 = 290

Text, T : 3124182284769678922843256...
initialize t = 0

h : $10^3$ mod 997 = 3
T : 3 1 2 4 1 8 2 2 8 4 7 6 9 6 7 8 9 3 2 5 6
# For the next four steps we are using Horner’s rule
# (first for loop in the algorithm), t = (d × t + T[s]) mod Q
s = 1, t = (d × t + T[s]) mod Q = (10 × 0 + 3) mod 997 = 3 mod 997 = 3
s = 2, t = (10 × 3 + 1) mod 997 = 31
s = 3, t = (10 × 31 + 2) mod 997 = 312
s = 4, t = (10 × 312 + 4) mod 997 = 133
# Now we start Rabin-Karp shuffle using rolling hash to compute
# the next substring from the previous substring.
# t = (d(t - T[s]h)+T[s+m])mod Q
# h = $10^3$ mod 997 = 3
s = 5, ( 10 × (133 - 3 × 3) + 1) mod 997 = 244
s = 6, ( 10 × (244 - 1 × 3) + 8) mod 997 = 424
s = 7, ( 10 × (424 - 2 × 3) + 2) mod 997 = 194
s = 8, ( 10 × (194 - 4 × 3) + 2) mod 997 = 825
s = 9, ( 10 × (825 - 1 × 3) + 8) mod 997 = 252
s = 10, ( 10 × (252 - 8 × 3) + 4) mod 997 = 290
we found our match