Algoritmo di Rabin-Karp

```
P <- Pattern di lunghezza M
T <- Testo di lunghezza N
h() funzione di hash

hp <- h(P)
for i = 1 to N - M + 1
    hs <- h(T[i .. i + M - 1])
    if hs = hp
        if T[i .. i + M - 1] = P
            return i
return -1
```

Utilizzo l'hashing per velocizzare i confronti, calcolo l'hash del pattern e l'hash di ogni sottostringa grande quando il pattern all'interno nel testo, confrontandoli hash potrei trovare una collisione, quindi devo fare un controllo carattere per carattere.

L'hash nel loop viene calcolato ad ogni iterazione, quindi aggiunge un calcolo lineare sulla lunghezza di M ad ogni iterazione. Questo metodo può essere ottimizzato tramite un *rolling hash*, che partendo dall'hash precedentemente calcolato calcola quello nuovo in tempo costante.

Rolling hash

Un tipo molto comune di rolling hash è il Rabin fingerprint, in questo caso ne usiamo una versione modificata. Devo fare hashing di un testo, ma lo considero come un numero in una base pari alla dimensione dell'alfabeto, con un numero primo di modulo.

Ad ogni *aggiunta* all'hash tolgo il carattere più significativo, faccio uno shift a sinistra e aggiungo un nuovo carattere dal testo, che prenderà la posizione meno significativa. Un alfabeto di dimensione d e un modulo m, applicati ad una stringa x di 4 caratteri, danno il seguente hash:

$$h = x_4 d^4 + x_3 d^3 + x_2 d^2 + x_1 d^1 + x_0 d^0 \mod m$$

Per aggiungere un carattere y

$$h' = (h - x_4 d^4) \cdot d + y d^0 \mod m$$

In generale per un hash di dimensione L e un testo S, l'hash iniziale si calcola in questo modo:

$$h(S_1) = \sum_{i=1}^{L} S[i] d^{L-i} \mod m$$

Mentre l'hash del carattere successivo:

$$h(S_{t+1}) = \left(h(S_t) - S[t]d^{L-1}\right) + S[t+1]d^0 \mod m$$

In cui la notazione $h(S_k)$ indica l'hash della porzione di testo $S[1..n]$ da k ad $k + L$.

Ridurre le collisioni aumenta l'efficienza dell'algoritmo, perchè non diventa necessario controllare le stringhe carattere per carattere, nel caso peggiore abbiamo un'efficienza pari a $O(L + nL)$ in cui n è la dimensione di S, nel caso medio invece $O(L + n)$.