Algorisme de Knuth-Morris-Pratt (KMP)

Problema: donada una cadena de n símbols C, i un patró P de m símbols trobar la posició p on ocorre P dins de C.

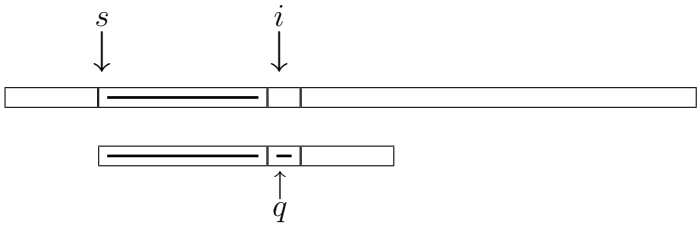
Algorisme trivial: Per a tota posició de C comparar els m símbols de P.

Millor cas: $\mathcal{O}(n)$ $\quad C = abababab\cdots aba^m,\ P = a^m$

Pitjor cas: $\mathcal{O}(nm)$ $\quad C = aaaa\cdots aab,\ P = a^{m-1}b$

Algorisme de Knuth-Morris-Pratt (KMP)

Notació:



s: posició del patró en C que està sent comprovada.

q: primera posició de P que no coincideix.

i: índex sobre C que es correspon amb q.

Es compleix que $C[s+1 \ldots i-1] = P[1 \ldots q-1]$

Algorisme de Knuth-Morris-Pratt (KMP)

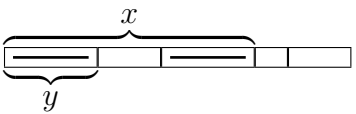
```
... aabaabaabaaa
    aabaaaa
       ^
```

Una vegada el prefix $aabaa$ del patrò ha sigut comparat amb èxit, i donat que el bloc aa apareix al principi i al final d'aquest, es pot deduir que la pròxima posició s que caldrà provar serà

```
... aabaabaabaaa
       aabaaaa
```

- No caldrà repetir la comprovació del primer bloc aa.
- L'índex que recorre C i compara amb P no torna enrere.

Algorisme de Knuth-Morris-Pratt (KMP)



x és el prefix de P comparat amb èxit.
y és el major prefix de x que és també sufix.

Funció prefix: $\pi : 1..m \longrightarrow 0..m-1$

Donat P, $\pi(j) =$ longitud del major prefix de P que és sufix propi de $P[1..j]$.

Així, quan es produesca una comparació errònia després d'haver comparat amb èxit j símbols del patró, sabrem que hem de desplaçar el patró $j - \pi(j)$ posicions a la dreta.

El pròxim símbol del patró que caldrà comparar serà el de la posició $\pi(j)+1$.

Algorisme de Knuth-Morris-Pratt (KMP)

Exemple:

patró	a	b	a	b	b
funció prefix	0	0	1	2	0

a	b	a	b	â	b	a	a
a	b	a	b	b̂			

Error en la posició 5 de P.
4 símbols comparats amb èxit.

a	b	a	b	â	b	a	a
		a	b	â	b	b	

La pròxima posició de P a comparar és:
$\pi(4) + 1 = 3$.

Algorisme de Knuth-Morris-Pratt (KMP)

Algorisme:

$i \leftarrow 1$ // posició sobre C

$q \leftarrow 0$ // índex sobre P en el rang $0..m-1$

mentre $(i \leq n) \wedge$ (queden símbols en C) fer

- mentre $(q > 0) \wedge (P[q+1] \neq C[i])$ fer $q \leftarrow \pi(q)$
- si $(P[q+1] = C[i])$ aleshores $q \leftarrow q+1$
- si $q = m$ aleshores PATRÓ TROBAT
- $i \leftarrow i+1$

Algorisme de Knuth-Morris-Pratt (KMP)

Podem imaginar que les cadenes són una estructura de dades i que el mentre i el si interns són "operacions".

Potencial: valor de la variable q (inicialment 0)

mentre: $\hat{c}_i = c_i + \Delta\phi = N - K \leq 0$

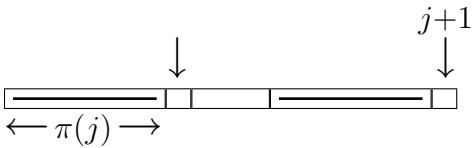
N és el nombre d'iteracions i K la quantitat en què decreix q que com a mínim serà 1 per iteració.

si: $\hat{c}_i = c_i + \Delta\phi = 1 + \max\{0, 1\} = 2$

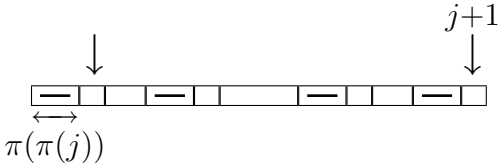
Si les dues "operacions" tenen cost amortitzat constant i el bucle extern fa de l'ordre de n operacions, el cost de l'algorisme és $\mathcal{O}(n)$.

Càlcul de la funció prefix

Suposem calculada $\pi(\ell)$ per a $1 \leq \ell \leq j$, i anem a calcular $\pi(j+1)$



si $P[j+1] = P[\pi(j)+1]$ aleshores
$$\pi(j+1) \leftarrow \pi(j)+1$$
sino ...



si $P[j+1] = P[\pi(\pi(j))+1]$ aleshores
$$\pi(j+1) \leftarrow \pi(\pi(j))+1$$
sino ...

si $\pi(\pi(\cdots\pi(j)\cdots)) = 0$ aleshores $\pi(j+1) \leftarrow (P[j+1] = P[1])$

Algorisme prefix

$\pi[1] \leftarrow 0$
$k \leftarrow 0$
per a $j \leftarrow 2$ fins a m fer
    mentre $(k > 0) \wedge (P[k+1] \neq P[j])$ fer $k \leftarrow \pi[k]$
    si $(P[k+1] = P[j])$ aleshores $k \leftarrow k+1$
    $\pi[j] \leftarrow k$

L'algorisme és estructuralment idèntic a l'anterior. Pel mateix motiu el cost serà lineal (en m).

El cost total de l'algorisme KMP és $\mathcal{O}(n+m)$.