Lezione 21 20/12/2023

LF-function

Proprietà di LF-mapping

Il suffisso che inizia con l'r-esimo simbolo σ della BWT B è l'r-esimo suffisso che inizia con il simbolo σ nel Suffix Array

LF-function

Funzione che fornisce la posizione lessicografica j del suffisso che inizia con il simbolo in posizione i della BWT, cioè B[i]

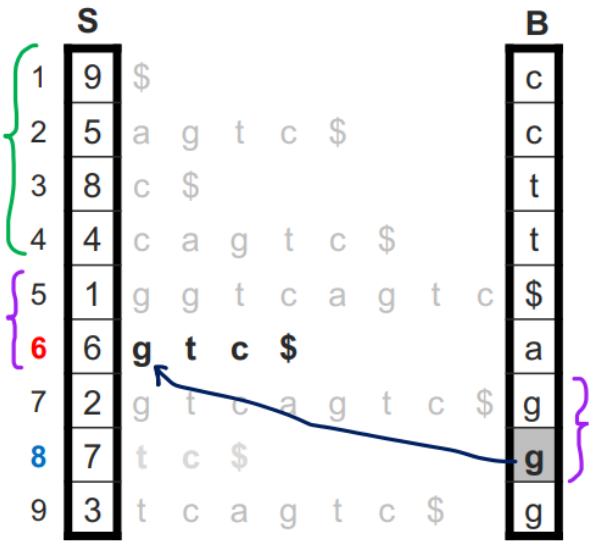
j = LF(i)

	1	2	3	4	5	6	7	8	9
T	g	g	t	c	a	g	t	c	$

Per i=8, B[i] = **g** è il simbolo iniziale del secondo suffisso che inizia con g, dal momento che B[i] è la seconda g in B

LF(i) deve restituire la posizione j=6 del suffisso **gtc$** nel Suffix Array

6 = 4 + 2



6 = C(B[8]) + Occ(8, B[8]) + 1

j = C(B[i]) + Occ(i, B[i]) + 1

LF-function

Ci sono 4 suffissi che iniziano con un simbolo inferiore di *g* (lo leggo dalla funzione B) e sommo 2 perché questa è la seconda *g* della BWT. Questo 2 lo leggo dalla funzione Occ (numero di prefissi fino alla posizione 7) + 1 perchè la Occ va fino ad una posizione prima.

Calcolo della backward extension

La backward extension di un Q-intervallo che inizia nella posizione b e finisce nella posizione e-1, aggiunge il simbolo sigma davanti e avrà una posizione di inizio e fine diversa.

$$\text{Q-intervallo } [b,e),\ \sigma \rightarrow \sigma\text{Q-intervallo } [b',e')$$

$$b' = LF(i_1)$$
$$e' = LF(i_k) + 1$$

$i_1 \rightarrow$ più piccola posizione in [b,e) tale che $B[i_1] = \sigma$
$i_k \rightarrow$ più grande posizione in [b,e) tale che $B[i_k] = \sigma$

LF-Function:

$$j = LF(i) = \mathbf{C}(B[i]) + \mathbf{Occ}(i, B[i]) + 1$$

$$b' = \mathbf{C}(B[i_1]) + \mathbf{Occ}(i_1, B[i_1]) + 1$$
$$e' = \mathbf{C}(B[i_k]) + \mathbf{Occ}(i_k, B[i_k]) + 1 + 1$$

$$b' = \mathbf{C}(\sigma) + \mathbf{Occ}(i_1, \sigma) + 1$$
$$e' = \mathbf{C}(\sigma) + \mathbf{Occ}(i_k, \sigma) + 1 + 1$$

In $B[b, i_1\text{-}1]$ **non** esistono simboli uguali a σ

$\Rightarrow$ il numero di simboli σ in $B[1, i_1\text{-}1]$ è uguale al numero di simboli σ in $B[1, b\text{-}1] \Rightarrow$ **Occ**(i_1, σ) = **Occ**(b, σ)

perchè i1 è quella più piccola che contiene sigma.

Quindi ci svincoliamo da i1, lo sostituiamo con b.

facciamo un ragionamento simile sul secondo

1. **Occ**(i_k, σ) = numero di simboli σ in $B[1, i_k\text{-}1]$
2. $B[i_k] = \sigma$

$\Rightarrow$ **Occ**$(i_k, \sigma) + 1 =$ **Occ**$(i_k\text{+}1, \sigma)$

In $B[i_k\text{+}1, e\text{-}1]$ non esistono simboli uguali a σ

$\Rightarrow$ il numero di simboli σ in $B[1, i_k]$ è uguale al numero di simboli σ in $B[1, e\text{-}1] \Rightarrow$ **Occ**$(i_k\text{+}1, \sigma) =$ **Occ**(e, σ)

~~i_1 → più piccola posizione in [b,e) tale che $B[i_1] = \sigma$~~

~~i_k → più grande posizione in [b,e) tale che $B[i_k] = \sigma$~~

$$b' = \mathbf{C}(\sigma) + \mathbf{Occ}(b, \sigma) + 1$$
$$e' = \mathbf{C}(\sigma) + \mathbf{Occ}(e, \sigma) + 1$$

e' = b' ⇒ il σQ-intervallo è vuoto

Procedura Backward_extend(b,e,σ)

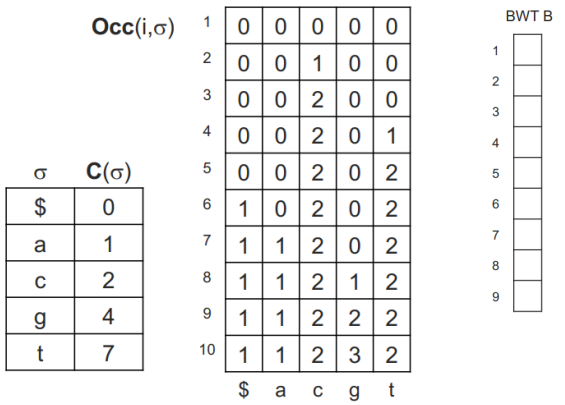
```
b' = C(σ) + Occ(b, σ) + 1
e' = C(σ) + Occ(e, σ) + 1

return [b', e')
```

Tempo O(1)

```
Procedura Search_pattern(P, S)
 begin
    n ← |S|
    [b,e) ← [1, n+1)
    i ← |P|
    while [b,e) is not null and i ≥ 1 do
      σ ← P[i]
      [b,e) ← Backward_extend(b,e,σ)
      i ← i-1
    if [b,e) is not null then
      output S[b], S[b+1], …, S[e-1]
 end
```

Complessità → O(m)

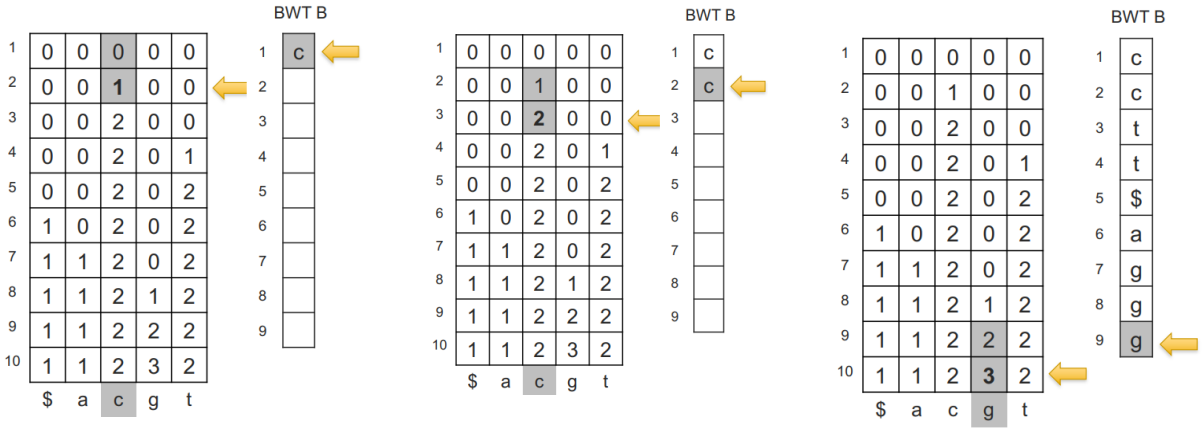
Esercizio FM-index è un self-index



σ	C(σ)
$	0
a	1
c	2
g	4
t	7

	$	a	c	g	t
1	0	0	0	0	0
2	0	0	1	0	0
3	0	0	2	0	0
4	0	0	2	0	1
5	0	0	2	0	2
6	1	0	2	0	2
7	1	1	2	0	2
8	1	1	2	1	2
9	1	1	2	2	2
10	1	1	2	3	2

Ricostruisci la BWT, che sarà lunga 1 in meno rispetto alla Occ.



Vediamo cos'è cambiato da ogni riga rispetto alla riga precedente.

Esercizio 1

Si consideri la seguente funzione **C**. Dire se il b-intervallo (per Q=b) è vuoto. Se non è vuoto, specificare le posizioni di inizio e fine.

σ	C(σ)
$	0
a	1
b	3
c	7
d	7

Funzione C

σ	C(σ)
$	0
a	1
b	**3**
c	**7**
d	7

Funzione C

C(c) – **C(b)** > 0 e quindi esistono suffissi che iniziano con il simbolo b.

→ b-intervallo **non** è vuoto

σ	C(σ)
$	0
a	1
b	**3**
c	**7**
d	7

Funzione C

Numero di suffissi che iniziano con σ < b → **C(b)** = **3**

Posizione nel Suffix Array del primo suffisso che inizia con b → **C(b)** + 1 = 4

Numero di suffissi che iniziano con σ < c → **C(c)** = **7**

Posizione nel Suffix Array dell'ultimo suffisso che inizia con b → **C(c)** = **7**

b-intervallo → [4,8)

Esercizio 2

Data la Burrows-Wheeler Transform B = accgt$ac di un testo T, si richiede di specificare l'FM-index supponendo Σ = {a,c,g,t}. Calcolare poi tramite FM-index la posizione j nel Suffix Array del suffisso che inizia con il terzo simbolo della BWT.

Funzione **C**?

B
a
c
c
g
t
$
a
c

σ	C(σ)
$	0

B
a
c
c
g
t
$
a
c

σ	C(σ)
$	0
a	1

B
a
c
c
g
t
$
a
c

σ	C(σ)
$	0
a	1
c	3

B
a
c
c
g
t
$
a
c

σ	C(σ)
$	0
a	1
c	3
g	6

B
a
c
c
g
t
$
a
c

σ	C(σ)
$	0
a	1
c	3
g	6
t	7

Funzione **Occ**?

B
a
c
c
g
t
$
a
c

σ	C(σ)
$	0
a	1
c	3
g	6
t	7

	$	a	c	g	t
1	0	0	0	0	0
2					
3					
4					
5					
6					
7					
8					
9					

B
a
c
c
g
t
$
a
c

σ	C(σ)
$	0
a	1
c	3
g	6
t	7

	$	a	c	g	t
1	0	0	0	0	0
2	0	1	0	0	0
3	0	1	1	0	0
4	0	1	2	0	0
5	0	1	2	1	0
6	0	1	2	1	1
7	1	1	2	1	1
8	1	2	2	1	1
9	1	2	3	1	1

Posizione j nel SA del suffisso che inizia B[**3**]?

	B
	a
	c
3 →	**c**
	g
	t
	$
	a
	c

σ	C(σ)
$	0
a	1
c	**3**
g	6
t	7

	$	a	**c**	g	t
1	0	0	0	0	0
2	0	1	0	0	0
3	0	1	**1**	0	0
4	0	1	2	0	0
5	0	1	2	1	0
6	0	1	2	1	1
7	1	1	2	1	1
8	1	2	2	1	1
9	1	2	3	1	1

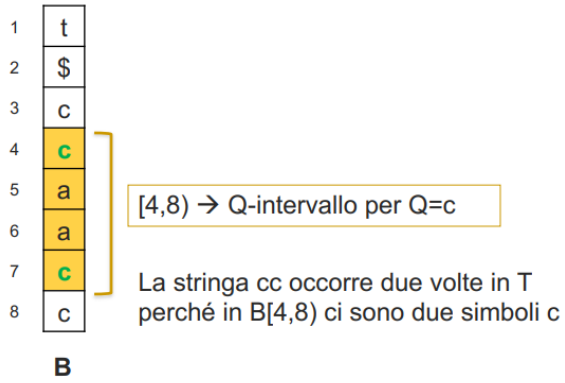
j = **C**(c) + **Occ**(**3**, c) + 1 = 3 + 1 + 1 = 5

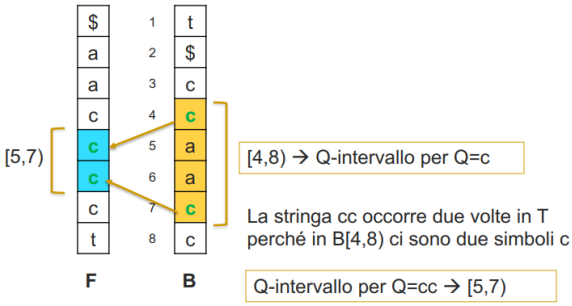
Esercizio 3

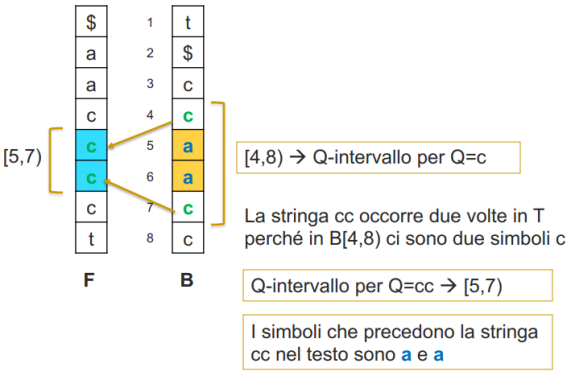
Data la BWT B = t$ccaacc di un testo T e sapendo che il Q-intervallo [4,8) rappresenta la stringa Q=c, specificare utilizzando unicamente B:

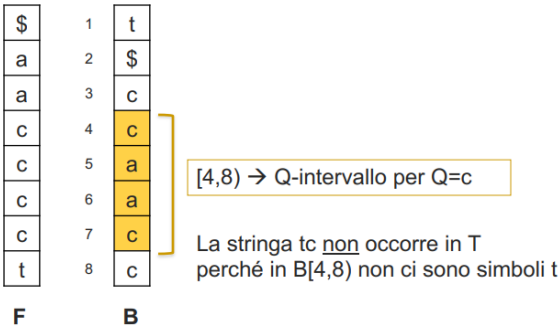
1) quante volte la stringa cc occorre in T, e specificare il Q-intervallo per Q=cc
2) quante volte la stringa tc occorre in T, e specificare il Q-intervallo per Q=tc

Indicare inoltre quali sono i simboli che nel testo precedono le occorrenze di cc e le occorrenze di tc.









Esercizio 4

Per un testo $-terminato di lunghezza n=10 la funzione **C** di FM-index è:

C($) = 0; **C**(a) = 1; **C**(c) = 4; **C**(g) = 6; **C**(t) = 6

1) Dire se una funzione **Occ** che per i = 11 vale:

Occ(11, $) = 1; **Occ**(11, a) = 2; **Occ**(11, c) = 3; **Occ**(11, g) = 2; **Occ**(11, t) = 2

è compatible con la funzione **C**. In caso contrario, specificare tutte le incongruenze.

2) Dire inoltre (in base alla funzione **C**) quanti sono i simboli g e quanti sono i simboli t nel testo.

RISPOSTA1: NO, perché

Distribuzione dei simboli derivata dalla funzione C:

C(a) - **C**($) = 1 simbolo $

C(c) - **C**(a) = 3 simboli a

C(g) - **C**(c) = 2 simboli c

C(t) - **C**(g) = 0 simboli g

10 - **C**(t) = 4 simboli t

che è diversa da quella specificata dalla funzione **Occ**

RISPOSTA2:

Numero di simboli g → 0

Numero di simboli t → 4

RISPOSTA1: NO, perché

a) **Occ**(11, $) + **Occ**(11, a) = 3 è diverso da **C**(c) = 4

b) **Occ**(11, $) + **Occ**(11, a) + **Occ**(11, c) + **Occ**(11, g) = 8 è diverso da **C**(t) = 6

RISPOSTA2:

Numero di simboli g → **C**(t) - **C**(g) = 0

Numero di simboli t → n - **C**(t) = 10 - 6 = 4

Esercizio 5

	$	a	c	g	t
1	0	0	0	0	0
2	0	0	0	0	1
3	0	0	0	1	1
4	0	1	0	1	1
5	0	1	0	2	1
6	0	1	0	2	2
7	0	2	0	2	2
8	0	2	1	2	2
9	1	2	1	2	2
10	1	2	1	3	2
11	1	2	1	4	2

Data la funzione **Occ** di FM-index di un testo T, derivare:

- la funzione **C**

- il simbolo della BWT in posizione 4

	$	a	c	g	t
7	0	2	0	2	2
8	0	2	1	2	2
9	1	2	1	2	2
10	1	2	1	3	2
11	1	2	1	4	2

σ	C(σ)
$	0
a	1
c	
g	
t	

Funzione **C**

	$	a	c	g	t
7	0	2	0	2	2
8	0	2	1	2	2
9	1	2	1	2	2
10	1	2	1	3	2
11	1	2	1	4	2

σ	C(σ)
$	0
a	1
c	3
g	
t	

Funzione **C**

	$	a	c	g	t
7	0	2	0	2	2
8	0	2	1	2	2
9	1	2	1	2	2
10	1	2	1	3	2
11	1	2	1	4	2

σ	C(σ)
$	0
a	1
c	3
g	4
t	

Funzione **C**

	$	a	c	g	t
7	0	2	0	2	2
8	0	2	1	2	2
9	1	2	1	2	2
10	1	2	1	3	2
11	1	2	1	4	2

σ	C(σ)
$	0
a	1
c	3
g	4
t	8

Funzione **C**

1	0	0	0	0	0
2	0	0	0	0	1
3	0	0	0	1	1
4	0	1	0	1	1
5	0	1	0	2	1
6	0	1	0	2	2
7	0	2	0	2	2
8	0	2	1	2	2
9	1	2	1	2	2
10	1	2	1	3	2
11	1	2	1	4	2
	$	a	c	g	t

Data la funzione **Occ** di FM-index di un testo T, derivare:

- la funzione **C**

- il simbolo della BWT in posizione 4

σ	**C**(σ)
$	0
a	1
c	3
g	4
t	8

Funzione **C**

B[4] = g

Inoltre, sapendo che [2,4) è il Q-intervallo per Q=a, dire sulla base di FM-index se la stringa aa occorre nel testo T.

La stringa aa occorre in T se in B[2,4) esiste almeno un simbolo a

In posizione 2 esiste un simbolo a?

1	0	0	0	0	0
2	0	0	0	0	1
3	0	0	0	1	1
4	0	1	0	1	1
5	0	1	0	2	1
6	0	1	0	2	2
7	0	2	0	2	2
8	0	2	1	2	2
9	1	2	1	2	2
10	1	2	1	3	2
11	1	2	1	4	2
	$	a	c	g	t

Inoltre, sapendo che [2,4) è il Q-intervallo per Q=a, dire sulla base di FM-index se la stringa aa occorre nel testo T.

σ	**C**(σ)
$	0
a	1
c	3
g	4
t	8

Funzione **C**

La stringa aa occorre in T se in B[2,4) esiste almeno un simbolo a

In posizione 2 esiste un simbolo a?
NO, B[2] è uguale a g

1	0	0	0	0	0
2	0	0	0	0	1
3	0	0	0	1	1
4	0	1	0	1	1
5	0	1	0	2	1
6	0	1	0	2	2
7	0	2	0	2	2
8	0	2	1	2	2
9	1	2	1	2	2
10	1	2	1	3	2
11	1	2	1	4	2
	$	a	c	g	t

Inoltre, sapendo che [2,4) è il Q-intervallo per Q=a, dire sulla base di FM-index se la stringa aa occorre nel testo T.

σ	**C**(σ)
$	0
a	1
c	3
g	4
t	8

Funzione **C**

La stringa aa occorre in T se in B[2,4) esiste almeno un simbolo a

In posizione 3 esiste un simbolo a?
YES, B[3] è uguale ad a

La stringa aa occorre in T

Esercizio 6

Si consideri la seguente funzione **Occ** di FM-index e si determini il testo originale

	$	a	c	g	t
1	0	0	0	0	0
2	0	0	0	0	1
3	1	0	0	0	1
4	1	0	1	0	1
5	1	0	2	0	1
6	1	1	2	0	1
7	1	2	2	0	1
8	1	2	3	0	1
9	1	2	4	0	1

BWT B?

	$	a	c	g	t
1	0	0	0	0	0
2	0	0	0	0	1
3	1	0	0	0	1
4	1	0	1	0	1
5	1	0	2	0	1
6	1	1	2	0	1
7	1	2	2	0	1
8	1	2	3	0	1
9	1	2	4	0	1

B
t

	$	a	c	g	t
1	0	0	0	0	0
2	0	0	0	0	1
3	1	0	0	0	1
4	1	0	1	0	1
5	1	0	2	0	1
6	1	1	2	0	1
7	1	2	2	0	1
8	1	2	3	0	1
9	1	2	4	0	1

B
t
$

	$	a	c	g	t
1	0	0	0	0	0
2	0	0	0	0	1
3	1	0	0	0	1
4	1	0	1	0	1
5	1	0	2	0	1
6	1	1	2	0	1
7	1	2	2	0	1
8	1	2	3	0	1
9	1	2	4	0	1

B
t
$
c

Ricostruzione di T

	$	a	c	g	t
1	0	0	0	0	0
2	0	0	0	0	1
3	1	0	0	0	1
4	1	0	1	0	1
5	1	0	2	0	1
6	1	1	2	0	1
7	1	2	2	0	1
8	1	2	3	0	1
9	1	2	4	0	1

F	B
$	t
a	$
a	c
c	c
c	a
c	a
c	c
t	c

T = accacct$