

ALGORITMI PER INTERNET E WEB: Indicizzazione e ricerca nei testi
 Compitino del 30 Maggio 2005

Esercizio 1 (15 punti):

Una stringa X si dice palindroma se $X = X^R$ (dove X^R denota la stringa X con i simboli in ordine inverso; per esempio, se $X = ABAC$ allora $X^R = CABA$). Un esempio di palindroma e' $X = ANNA$. Un testo T si dice che contiene un palindromo di ordine k , per un intero $k > 1$, se esiste una sottostringa X di T tale che

- i) X e' palindroma,
- ii) la sua lunghezza e' $|X| = k$.

Per esempio, $T = ABACACC$ contiene almeno un palindromo di ordine $k = 3$. Dato un testo T , si supponga di avere a disposizione l'algoritmo di costruzione lineare di un suffix tree S . Descrivere un algoritmo che, presi in ingresso T , S e k , determina se T contiene un palindromo di ordine k .

[Suggerimento: costruire S su $T\$T^R\#$, dove $\$$ e $\#$ sono due caratteri speciali di fine stringa.]

Soluzione ottima:

Sia X una palindroma di ordine k . Prendiamo il caso k pari (per k dispari e' una lieve variante). Possiamo scrivere X come la concatenazione delle sue meta': $X = X[1..k/2] X[k/2+1..k]$. Deve valere $X[1..k/2] = X[k/2+1..k]^R$.

Sia n la lunghezza $|T|$ del testo. Siccome X e' palindroma, $X[1..k/2]$ appare sia in T , in posizione i , che in T^R , in posizione $n - (i+k-1) + 1$ per un certo valore i . Se prendiamo la concatenazione $T\$T^R\#$, l'occorrenza sara' in posizione i e in posizione $i' = (n+1) + [n - (i+k-1) + 1]$. Si veda pag. 7 delle dispense GI98].

Prendiamo il suffix tree S costruito sulla concatenazione $T\$T^R\#$. Dobbiamo considerare gli archi di S in cui troviamo le sottostringhe di lunghezza $k/2$. Notare che ci possono essere al piu' n di tali archi. Si noti che tali archi sottendono insiemi disgiunti di foglie discendenti in S .

Fra questi archi, uno deve corrispondere a $X[1..k/2]$. Per individuarlo, basta esaminare le sue foglie discendenti. Tra tali foglie, una deve contenere la posizione i e un'altra la posizione i' , per un qualche valore di i . Un semplice vettore di appoggio di $O(n)$ elementi permette di verificare se tali posizioni i e i' esistono, con costo totale di $O(n)$ tempo per tutti gli archi cosi' considerati.

Esercizio 2 (15 punti):

Scrivere l'automa di Knuth-Morris-Pratt semplificato per il pattern $P = \text{ATAATATAATAT}$.

Soluzione:

Basta scrivere la tabella Φ :

i	0	1	2	3	4	5	6	7	8	9	10	11	12
$\Phi[i]$	-1	0	0	1	1	2	3	2	3	4	5	6	7

