Palindrome

4 dicembre 2000

Indichiamo con $f(w)$ il minimo numero di caratteri necessari per rendere la stringa w palindroma. Useremo le seguenti notazioni:

- $|w|$ indica la lunghezza della parola w;
- w^R indica il "reversal" di w, cioè la parola che si ottiene leggendo w al contrario; notate che $(wv)^R = v^R w^R$.

Conviene fare alcune osservazioni:

1. $f(w) = 0$ se $|w| \leq 1$ (una stringa di 0 o 1 carattere è già palindroma);
2. se w è palindroma, e $w = \alpha v \beta$, allora o α è un prefisso di β^R, o β è un prefisso di α^R, e inoltre, togliendo dalle due estremità della stringa il prefisso comune si ottiene di nuovo una stringa palindroma.

Mostriamo due lemmi:

Lemma 1 Se a è un carattere, $f(awa) = f(w)$.

Dimostrazione. Si noti, intanto, che per rendere palindroma la stringa awa si può semplicemente rendere palindroma w, quindi $f(awa) \leq f(w)$. Una generica stringa ottenuta inserendo caratteri in awa ha la forma $\alpha a w' a \beta$, dove w' è ottenuta inserendo dei caratteri in w. Supponiamo che la stringa $v = \alpha a w' a \beta$ sia palindroma e ottenuta aggiungendo il minor numero di caratteri possibili in awa, cioè

$$|\alpha| + |\beta| + |w'| - |w| = f(awa). \tag{1}$$

Per le osservazioni fatte, o α è un prefisso di β^R o viceversa. Supponiamo che si verifichi il primo caso (l'altro è analogo): allora $v = \alpha a w' a \gamma \alpha^R$ e $aw'a\gamma$ è palindroma. Poiché $aw'a\gamma$ contiene awa ed è palindroma, i caratteri aggiunti saranno almeno $f(awa)$, cioè

$$|w'| - |w| + |\gamma| \geq f(awa). \tag{2}$$

Mettendo insieme (1) e (2), si ottiene

$$f(awa) = 2|\alpha| + |\gamma| + |w'| - |w| \geq 2|\alpha| + f(awa)$$

da cui segue che α deve essere la stringa vuota. Ma allora $v = aw'a\beta$ e a è prefisso di β^R o β è prefisso di a. I casi possibili sono solo i seguenti:

- β è vuota: allora $v = aw'a$ e w' è palindroma (per cui $|w'| - |w| = f(w)$) e il risultato segue da (1);
- $\beta = \gamma a$, per qualche γ. Allora $v = aw'\gamma a$ e $w'\gamma$ è palindroma. Da questo segue che $|w'| - |w| + |\gamma| \geq f(w)$ e quindi, usando (1) $f(awa) \geq 1 + f(w)$. Ciò è impossibile, poiché $f(awa) \leq f(w)$.

∎

Lemma 2 Se a e b sono due caratteri diversi, $f(awb) = 1 + \min\{f(aw), f(wb)\}$.

Dimostrazione. Si noti, intanto, che per rendere palindroma la stringa awb si può procedere o rendendo palindroma aw e aggiungendo un b prima e dopo (totale caratteri aggiunti: $f(aw)+1$) o rendendo palindroma wb e aggiungendo un a prima e dopo (totale caratteri aggiunti: $f(wb)+1$). Quindi $f(awb) \leq 1+\min\{f(aw), f(wb)\}$. Il resto della dimostrazione procede come nel lemma precedente. ∎

Ovviamente, il modo teoricamente più semplice per calcolare f sarebbe a questo punto scrivere una procedura ricorsiva che consideri i casi precedenti. Tale soluzione non è però abbastanza efficiente (il numero di chiamate ricorsive cresce esponenzialmente con la lunghezza della stringa).

Si può notare, d'altra parte, che il calcolo di f su una stringa comporta la valutazione di f su una sua sottostringa opportuna. Potremmo quindi pensare di calcolare f su tutte le sottostringhe della stringa data, utilizzando la programmazione dinamica.

In particolare, se $w = a_0 a_1 \ldots a_{n-1}$ è la stringa in input, di lunghezza n, consideriamo una matrice triangolare $M[i][k]$ (con k che varia tra 0 e n, e i che varia tra 0 e $n-k$) in cui $M[i][k]$

corrisponde a $f(a_i a_{i+1} \dots a_{i+k})$. Tale matrice si può facilmente riempire a partire da $k = 0$ facendo salire k.

Purtroppo la matrice M ha dimensione $5000 * 5000/2$: decisamente eccessiva! D'altro canto si può notare che per riempire la riga $M[-][k]$ occorre solo conoscere le righe $M[-][k-1]$ e $M[-][k-2]$. Questo è l'approccio adottato nell'algoritmo.