



Esame di ALGORITMI: Corso di Laurea in Informatica
9 febbraio 2022

Problema

Una string S è un interleaving di due stringhe X , Y se S è formata da tutti e soli i caratteri di X e Y e l'ordine dei caratteri è preservato. Per esempio

ADEBCF e ABDECF sono entrambe interleaving di ABC e DEF,

ABAEFC e AAEBFC sono entrambe interleaving di ABC e AEF,

ABECF e AACEFB non sono interleaving di ABC e AEF

Si proponga un algoritmo per stabilire se una stringa S è un interleaving di X e Y e se ne studi la complessità.

Programmazione dinamica

Si controlla ogni carattere della stringa S con ogni carattere delle stringhe X e Y .

Se i caratteri di X e Y sono diversi, allora si riesce a scegliere immediatamente quale sottostringa appartiene a S .

Se ad esempio il primo carattere di X e il primo di Y sono uguali e si trova lo stesso in S non si sa se il carattere in S appartiene ad X o ad Y

S
A B A E F C
| | X X
A B C A E F
x y

S
A B A E F C
 ?
A B C A E F
x y

Sbagliato perchè B non è il primo carattere nè di X nè di Y

Si potrebbero provare entrambe le possibilità che è la base della programmazione dinamica

```
// S è la stringa da controllare
// x è la prima sottostringa
// y è la seconda sottostringa
// i è la lunghezza della stringa x
// j è la lunghezza della stringa y
// Array indicizzati da 1
is_interlieved(S, x, y, i, j):
    if i == 0 && j == 0
        return true
    if i + j != len(S)
        return false

    x_char = i > 0 ? x[i] : null
    y_char = j > 0 ? y[j] : null
    S_char = S[i + j]

    if S_char == x_char && S_char != y_char
        return is_interlieved(S, x, y, i--, j)
    else if S_char == y_char && S_char != x_char
        return is_interlieved(S, x, y, i, j--)

    if S_char == x_char && S_char == y_char
        return is_interlieved(S, x, y, i--, j) || is_interlieved(S, x, y, i, j--)

    return false
```

Questo algoritmo risolve il problema, ma la complessità è troppo elevata, per ridurla bisogna implementare la memoizzazione.