

Chiarimento sul calcolo esatto delle somme nelle serie

Dato l'algoritmo:

```
COMPUTE(a,b)
n <- length(a)
m = +infinito
for i=1 to n-1 do
  C[i,n-1] <- a_i
  m <- MIN(m,C[i,n-1])
for j=0 to n-1 do
  C[0,j] <- b_j
  m <- MIN(m,C[0,j])
for i=1 to n-2 do
  for j=n-2 downto i do
    C[i,j] <- C[i-1,j-1] * C[i,j+1]
    m <- MIN(m,C[i,j])
return m
```

La somma che calcola esattamente le somme è quella sotto presentata.

Analizziamo meglio come si ottiene il risultato. In particolare si nota che “j” è un indice che arriva fino ad “i” partendo da “n-2”. Si nota come infatti, al massimo possa fare “n-1” iterazioni.

Quindi in ogni operazione si trasla l'indice della sommatoria buttandolo nel risultato.

Spiegato passo passo sarebbe quindi:

- la somma iniziale vale 1 in quanto i nostri elementi non sono inizializzati ad alcun valore e hanno distanza massima di 1
 - dato che j va da i ad n-1 iterazioni in n-2, nel passaggio dopo scrivo “n-1-i”.
 - a questo punto traslo “n-2” e lo butto dentro al risultato.
 - sto facendo questa operazione due volte, quindi divido per due.
- Il risultato è quanto segue:

$$T(n) = \sum_{i=1}^{n-2} \sum_{j=i}^{n-2} 1 = \sum_{i=1}^{n-2} n - 1 - i = \sum_{k=1}^{n-2} k = (n-1)(n-2)/2.$$

A titolo di esempio trattiamo anche il caso dell'algoritmo:

```
INIT_M(n)
if n=1 then return 1
if n=2 then return 2
for i=1 to n-1 do
  M[i,i] = 1
  M[i,i+1] = 2
M[n,n] = 1
for i=1 to n-2 do
  for j=i+2 to n do
    M[i,j] = 0
return REC_M(1,n)

REC_M(i,j)
if M[i,j] = 0 then
  M[i,j] = REC_M(i+1,j-1) * REC_M(i+1,j) * REC_M(i,j-1)
return M[i,j]
```

Il risultato è:

$$T(n) = \sum_{i=1}^{n-2} \sum_{j=i+2}^n 2 = 2 \sum_{i=1}^{n-2} n - i - 1 = 2 \sum_{k=1}^{n-2} k = (n-2)(n-1)$$

Di nuovo:

- la somma iniziale vale 2 in quanto i nostri elementi sono inizializzati ad 1 e a 2 e, in particolare, sono distanti massimo 2 nella scrittura dei cicli dell'algoritmo
- come prima, j è dato da "n-1-i" confronti ed iterazioni, questo perché j è maggiore di i come indice quindi avrà al massimo quel costo.
- sapendo poi che appunto entrambi tendono a "n-2" perché si vede dall'ultimo ciclo, massime operazioni che entrambi fanno, nell'ultimo passaggio mettiamo ad esponente comune "n-2" e come prima si trasla nel risultato "n-1" dell'indice j

Da questo si ha il risultato.