ESERCIZIO DA ESAME 21/02/23

Esercizio 2 (9 punti) Per $n > 0$, siano dati due vettori a componenti intere $\mathbf{a}, \mathbf{b} \in \mathbf{Z}^n$. Si consideri la quantità $c(i,j)$, con $0 \le i \le j \le n-1$, definita come segue:

$$c(i,j) = \begin{cases} a_i & \text{se } 0 < i \le n-1 \text{ e } j = n-1, \\ b_j & \text{se } i = 0 \text{ e } 0 \le j \le n-1, \\ c(i-1,j-1) \cdot c(i,j+1) & 0 < i \le j < n-1. \end{cases}$$

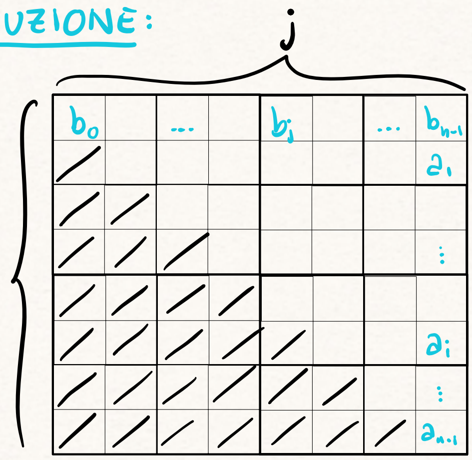
Si vuole calcolare la quantità $m = \max\{c(i,j) : 0 \le i \le j \le n-1\}$.

(a) Fornire un algoritmo iterativo bottom-up per il calcolo di m.

(b) Valutare la complessità *esatta* dell'algoritmo, associando costo unitario alle moltiplicazioni tra numeri interi e costo nullo a tutte le altre operazioni.

SOLUZIONE:

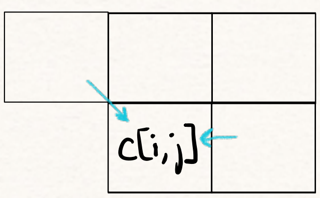
a)



$c(i,j)$ per $0 \le i \le j \le n-1$

INIZIALIZZAZIONE: $c(i, n-1) = a_i$ $i \ge 1$
$c(0,j) = b_j$ $j \ge 0$
$m = -\infty$

ESPLORAZIONE:



$c(i,j) = c(i-1,j-1) \cdot c(i,j+1)$ se $0 < i \le j < n-1$

esploro righe in senso crescente

esploro colonne in senso decrescente, $j \ge i$

```
Compute_m(a,b):
    n = len(a)
    m = -infty
    allocate C[1...n-1, 1...n-1]
    for i=1 to n-1:
        C[i,n-1] = a[i]
        m = max(m, C[i,n-1]) // primo update del max
```

```
for j=0 to n-1:
    C[0,j] = b[j]
    m = max(m, C[0,j]) // update del max
for i=1 to n-2:
    for j=n-2 to i:
        C[i,j] = C[i-1,j-1] * C[i,j+1]
        m = max(m, C[i,j])
return m
```

b) Compute_m(a,b):

n = len(a) → $c=0$

m = -infty → $c=0$

allocate C[1...n-1, 1...n-1] → $c=0$

$= 0$

for i=1 to n-1: → $\sum_{i=1}^{n-1}$

  C[i,n-1] = a[i] → $c=0$

  m = max(m, C[i,n-1]) → $c=0$

$= 0$

for j=0 to n-1: → $\sum_{j=0}^{n-1}$

  C[0,j] = b[j] → $c=0$

  m = max(m, C[0,j]) → $c=0$

$= 0$

for i=1 to n-2: → $\sum_{i=1}^{n-2}$

  for j=n-2 to i: → $\sum_{i=1}^{n-2}$

    C[i,j] = C[i-1,j-1] * C[i,j+1] → $c=1$

    m = max(m, C[i,j]) → $c=0$

$\sum_{i=1}^{n-2}\sum_{j=1}^{n-2} 1$

return m → $c=0$

$$T(n) = \sum_{i=1}^{n-2}\sum_{j=i}^{n-2} 1 = \sum_{i=1}^{n-2} \underbrace{n-1-i}_{K} = \sum_{K=1}^{n-2} K = \frac{(n-1)(n-2)}{2}$$

$i=1 \leadsto K=n-2$

$i=n-2 \leadsto K=n-1-n+2=1$

ESERCIZIO DA ESAME 06/07/21

Esercizio 2 (9 punti) Data una stringa $X = x_1, x_2, \ldots, x_n$, si consideri la seguente quantità $\ell(i,j)$, definita per $1 \leq i \leq j \leq n$:

$$\ell(i,j) = \begin{cases} 1 & \text{se } i = j \\ 2 & \text{se } i = j-1 \\ 2 + \ell(i+1, j-1) & \text{se } (i < j-1) \text{ e } (x_i = x_j) \\ \sum_{k=i}^{j-1} (\ell(i,k) + \ell(k+1,j)) & \text{se } (i < j-1) \text{ e } (x_i \neq x_j). \end{cases}$$

1. Scrivere una coppia di algoritmi INIT_L(X) e REC_L(X,i,j) per il calcolo memoizzato di $\ell(1,n)$.
2. Si determini la complessità *al caso migliore* $T_{\text{best}}(n)$, supponendo che le uniche operazioni di costo unitario e non nullo siano i confronti tra caratteri.

SOLUZIONE:

$$\ell(i,j) = \begin{cases} 1 & \text{se } i = j \\ 2 & \text{se } i = j-1 \\ 2 + \ell(i+1, j-1) & \text{se } (i < j-1) \text{ e } (x_i = x_j) \\ \sum_{k=i}^{j-1} (\ell(i,k) + \ell(k+1,j)) & \text{se } (i < j-1) \text{ e } (x_i \neq x_j). \end{cases}$$

CASI BASE ⇝ INIT_L

CASI RICORSIVI ⇝ REC_L

1.
```
INIT_L(X):
    n = len(X)
    if n = 1:
        return 1
    if n = 2:
        return 2
    allocate L[1...n, 1...n]
    for i=1 to n-1:      // per trattare j = i+1
        L[i,i] = 1
        L[i,i+1] = 2
    L[n,n] = 1
    for i=1 to n-2:
        for j=i+2 to n:
            L[i,j] = 0
    return REC_L(X,1,n)
```

abbrevia procedura per casi $X = x_1$, $X = x_2$

inizializza con valore di controllo

↪ chiama l'algoritmo ricorsivo

```
REC_L(X, i, j):
  if L[i,j] = 0:
    if X[i] = X[j]:
      L[i,j] = 2 + REC_L(X, i+1, j-1)
    else for k = i to j-1:
      L[i,j] = L[i,j] + REC_L(X, i, k) + REC_L(X, k+1, j)
  return L[i,j]
```

$$\ell(i,j) = \begin{cases} \dots \\ 2 + \ell(i+1, j-1) & \text{se } (i < j-1) \text{ e } (x_i = x_j) \\ \sum_{k=i}^{j-1} (\ell(i,k) + \ell(k+1,j)) & \text{se } (i < j-1) \text{ e } (x_i \neq x_j). \end{cases}$$

2. T_{best} ?

→ Caso migliore è se $X[i] = x \quad \forall i = 1 \dots n$, cioè tutti i caratteri sono uguali.

$L[i,j] = 2 + REC\_L(X, i+1, j-1) \rightarrow$
$$\left.\begin{aligned} L[1,n] &= 2 + L[2,n-1] \\ L[2,n-1] &= 2 + L[3,n-2] \\ &\vdots \end{aligned}\right\} \left\lfloor \frac{n}{2} \right\rfloor \text{ chiamate}$$

$\Rightarrow \quad T_{best}(n) = 2 + T_{best}(n-2) \qquad \Rightarrow \quad T_{best}(n) = O(n)$

$\square$

ESERCIZIO DA ESAME 08/09/23

Esercizio 2 (9 punti) Sia $n > 0$ un intero. Si consideri la seguente ricorrenza $M(i,j)$ definita su tutte le coppie (i,j) con $1 \le i \le j \le n$:

$$M(i,j) = \begin{cases} 1 & \text{se } i = j, \\ 2 & \text{se } j = i+1, \\ M(i+1,j-1) \cdot M(i+1,j) \cdot M(i,j-1) & \text{se } j > i+1. \end{cases}$$

1. Scrivere una coppia di algoritmi INIT_M(n) e REC_M(i,j) per il calcolo memoizzato di $M(1,n)$.
2. Calcolare il numero esatto $T(n)$ di moltiplicazioni tra interi eseguite per il calcolo di $M(1,n)$.

SOLUZIONE:

1.
```
INIT_M(n):
    if n = 1:
        return 1
    if n = 2:
        return 2                  abbrevia procedura per casi n=1, n=2
    allocate M[1...n, 1...n]
    for i=1 to n-1:  // per trattare j=i+1
        M[i,i] = 1
        M[i,i+1] = 2
    M[n,n] = 1
    for i=1 to n-2:
        for j=i+2 to n:           inizializza con valore di controllo
            M[i,j] = 0
    return REC_M(1,n)

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

$M(i,j) =$ $M(i+1,j-1) \cdot M(i+1,j) \cdot M(i,j-1)$ se $j > i+1$.

2. M[i,j] = REC_M(i+1,j-1) * REC_M(i+1,j) * REC_M(i,j-1) $\rightarrow c = 2$

$c = 2$ per $i = 1$ to $n-2$, $j = i+2$ to n ($j=i$ e $j=i+1$ casi base)

$$\Rightarrow T(n) = \sum_{i=1}^{n-2} \sum_{j=i+2}^{n} 2 = 2 \cdot \sum_{i=1}^{n-2} \underbrace{n-i-1}_{K \text{ (come es. 1)}} = 2 \cdot \sum_{K=1}^{n-2} K = (n-2) \cdot (n-1)$$