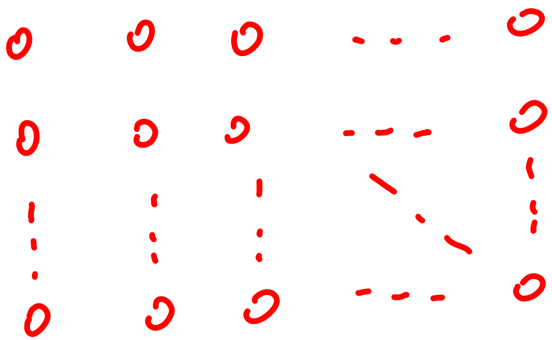
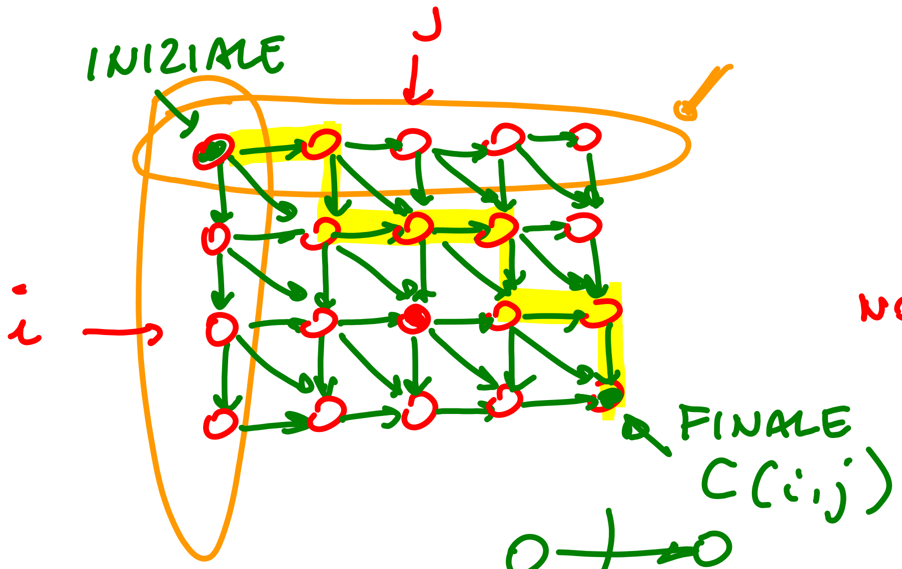
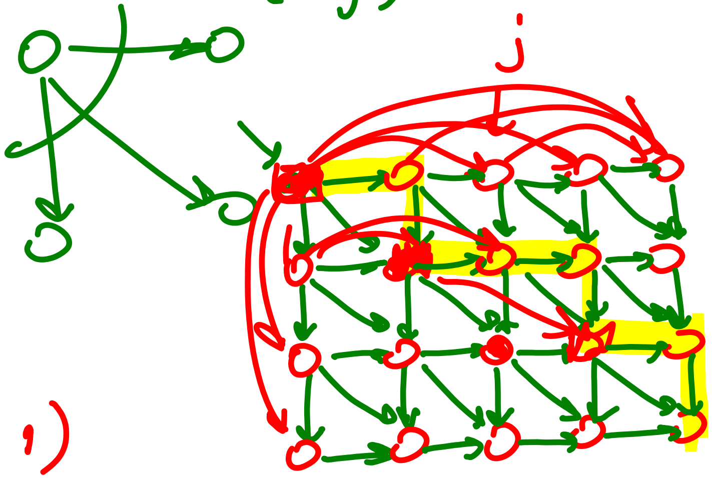
(a) Supponiamo di avere una griglia rettangolare di dimensioni $n \times m$ e indichiamo con (i, j) il nodo posizionato all'intersezione della i-esima riga e j-esima colonna, con $1 \leq i \leq n$ e $1 \leq j \leq m$.
Supponiamo inoltre che dal nodo (i, j) sia possibile raggiungere soltanto i nodi $(i, j+1)$, $(i+1, j)$ e $(i+1, j+1)$ (se presenti), in ciascun caso con il medesimo costo $\boxed{C(i, j)}$ dove C è una data matrice $n \times m$ di interi non negativi.
Utilizzando la metodologia della programmazione dinamica, si descriva un algoritmo per il calcolo del *costo minimo* di un percorso consentito dal nodo $(1, 1)$ al nodo (n, m) e lo si analizzi dal punto di vista della complessità computazionale.

(b) **(Facoltativo)** Si estenda il precedente algoritmo in modo da calcolare un *percorso di costo minimo* dal nodo $(1, 1)$ al nodo (n, m).





NODO (i,j)



$j=1$

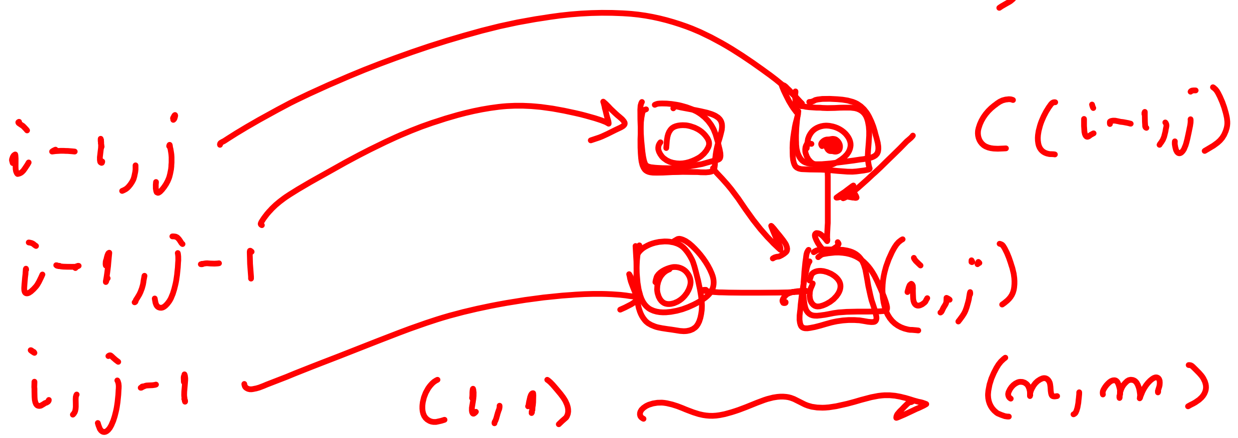
$$m[i,j] = m[i-1, 1] + C(i-1, 1)$$

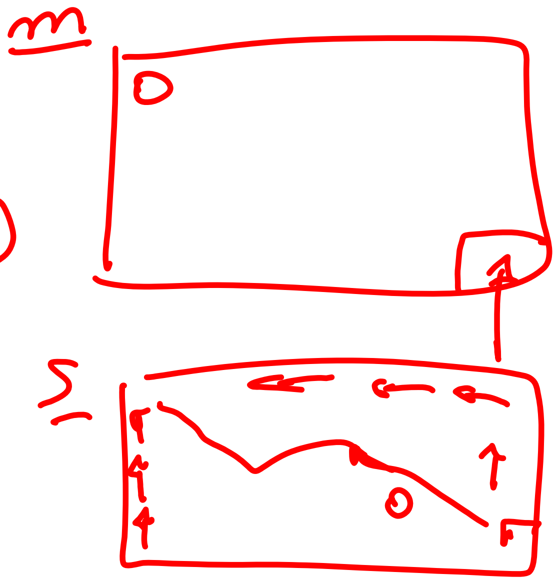
$m[i,j]$ = costo minimo di un percorso da $(1,1)$ a (i,j)

$$
m[i,j] = \begin{cases}
0 & i=1 \wedge j=1 \\
m[1,j-1] + C(1,j-1) & i=1 \wedge j>1 \\
m[i-1,1] + C(i-1,1) & i>1 \wedge j=1 \\
\min\{ m[i-1,j] + C(i-1,j),\ m[i-1,j-1] + C(i-1,j-1),\ m[i,j-1] + C(i,j-1) \} & i>1 \wedge j>1
\end{cases}
$$

$(1,1)$

$O(n \cdot m)$





Dato un array bidimensionale $a[1..2, 1..n]$ di interi relativi, con $n \geq 2$, utilizzando la metodologia della programmazione dinamica si progetti un algoritmo che calcoli n costanti $1 \leq c_i \leq 2$, con $i = 1, \ldots, n$, tali che

- $c_i + c_{i+1} \leq 3$, per $i = 1, \ldots, n-1$,
- $\sum_{i=1}^{n} a[c_i, i]$ sia massima,

e lo si analizzi dal punto di vista della complessità computazionale.

Si illustri inoltre l'algoritmo sul seguente array $a[1..2, 1..6]$:

$8 + 6 + 7 \cancel{-4} + 7 + \cancel{4} = 28$

i	$i+1$
1	1
1	2
2	1
~~2~~	~~2~~

a	1	2	3	4	5	6
1	2	6	5	-4	7	3
2	8	1	7	-5	3	4

	1	2	3	4	5	6
m_1	2	14	19	17	24	27
m_2	8	3	21	14	20	28

$m_1[i-1]$

$m_1[i]$ = valore max di una somma ammissibile che contiene l'elem. $a[1,i]$

$m_2[i]$ = valore max di una somma ammissibile che contiene l'elem. $a[2,i]$

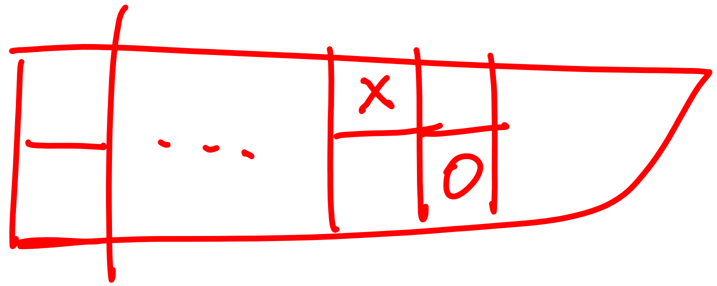
$\max(m_1[n], m_2[n])$

28

$$\boxed{\begin{array}{l} c_1 = 2 \\ c_2 = 1 \\ c_3 = 2 \\ c_4 = 1 \\ c_5 = 1 \\ c_6 = 2 \end{array}}$$

$$m_1[i] = \begin{cases} a[1,1] & i = 1 \\ \max(m_1[i-1], m_2[i-1]) + a[1,i] & i > 1 \end{cases}$$

$$m_2[i] = \begin{cases} a[2,1] & i = 1 \\ m_1[i-1] + a[2,i] & i > 1 \end{cases}$$



La celebre programmatrice Penelope C., chiamata a scrivere per conto della compagnia navale Ulysses un programma per il calcolo ottimale del prodotto di sequenze di matrici, per errore produce un algoritmo che effettua tali prodotti nel *peggior* modo possibile.

(a) Si definisca in maniera precisa il problema della moltiplicazione di sequenze di matrici.
(b) Utilizzando la metodologia della programmazione dinamica, si ricostruisca l'algoritmo trovato da Penelope C.
(c) Si stabilisca se esistono sequenze di matrici il cui prodotto viene calcolato in maniera ottimale anche dal programma di Penelope C.

$p \times p$

p^3

$$m[i,j] = \begin{cases} 0 & i = j \\ \max\limits_{i \le k < j} \left(m[i,k] + m[k+1,j] + p_{i-1} \cdot p_k \cdot p_j \right) & i < j \end{cases}$$

$$\overbrace{(A_i \times \dots \times A_k)}^{m[i,k]}\overbrace{(A_{k+1} \dots A_j)}^{m[k+1,j]}$$

$p_{i-1} \times p_k$ $\qquad$ $p_k \times p_j$

ESERCIZIO 1

Sia data una sequenza ordinata di n stringhe $w_1, w_2, \ldots, w_n$.

Si supponga che per concatenare due stringhe w e w' si incorra in un costo uguale a $(|w| + |w'|)^2$, dove $|\cdot|$ denota l'operatore lunghezza.

Si progetti un algoritmo, inquandrandolo nel contesto opportuno, per effettuare la concatenazione delle n stringhe date (così da ottenere la stringa $w_1 w_2 \cdots w_n$) che incorra nel costo complessivo minimo e se ne valuti la complessità computazionale.

$$m[i,j] = \begin{cases} 0 & i = j \\ \min\limits_{i \le k < j} \left(m[i,k] + m[k+1,j] + \left(\sum\limits_{\ell=i}^{j} |w_\ell| \right)^2 \right) & i < j \end{cases}$$

Sia $\mathcal{S}$ un sequenza finita di nodi e sia $label : \mathcal{S} \longrightarrow \mathbb{N}$ una data funzione definita sugli elementi della sequenza $\mathcal{S}$.
Si descriva un algoritmo che costruisca un albero binario ordinato **pieno** $\mathcal{T}$ tale che:

- la sequenza delle foglie di $\mathcal{T}$, lette da sinistra a destra, coincide con $\mathcal{S}$;
- il valore della funzione φ calcolato sulla radice di $\mathcal{T}$ sia **minimo**, dove:

$$\varphi(x) = \begin{cases} label(x) & \text{se } x \text{ è una foglia} \\ \varphi(\ell)^2 + \varphi(r)^2 & \text{altrimenti, con } \ell \text{ ed } r \text{ figlio sinistro e figlio destro di } x \text{, rispettivamente.} \end{cases}$$

$$(m[i,k])^2 + (m[k+1,j])$$

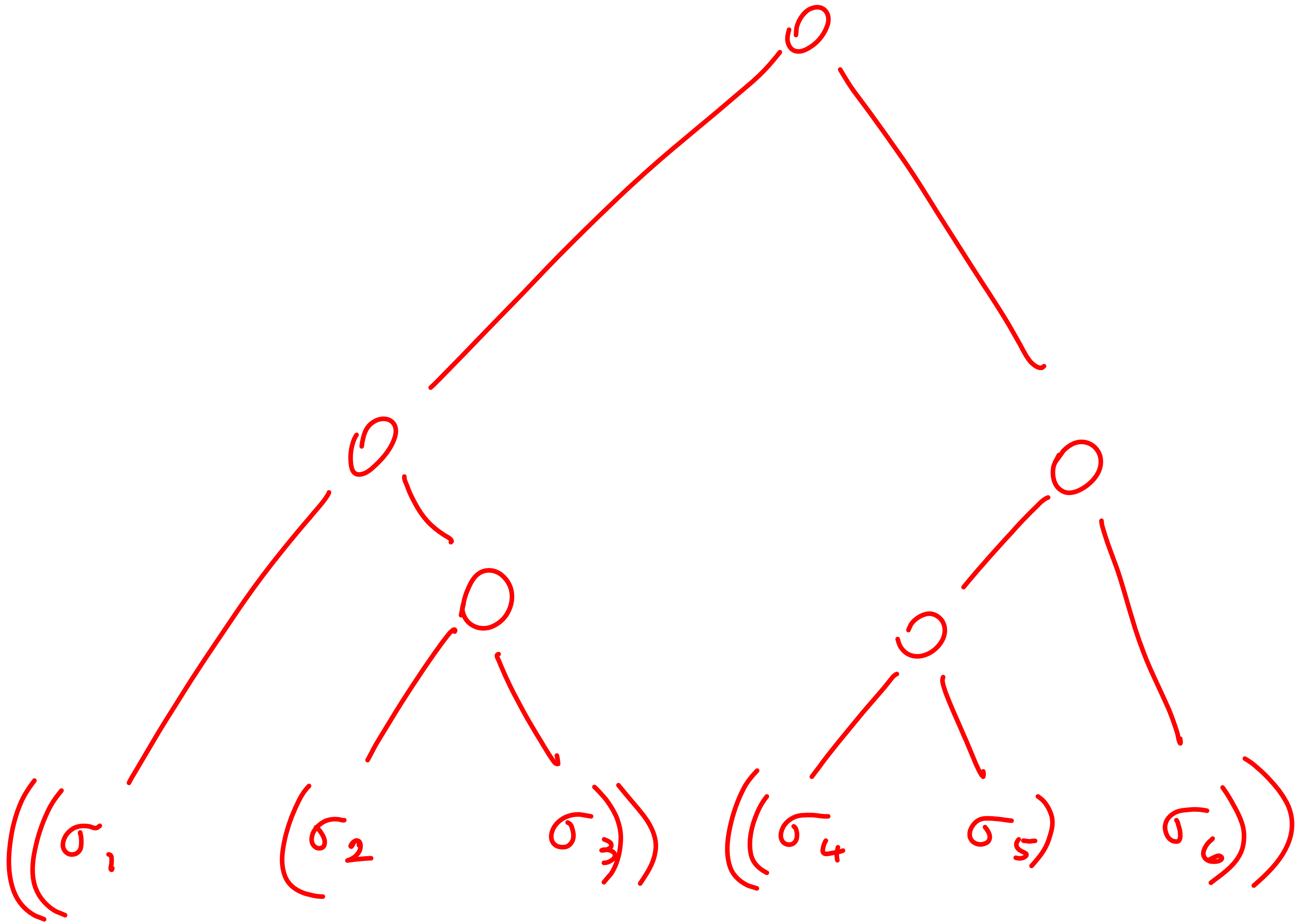
$$(\overset{i}{\circ} \quad \circ \quad \circ \quad \overset{k}{\circ})(\circ \quad \circ \quad \circ)$$

$$k := i, \dots, j-1$$

$$m[i,j] = \begin{cases} label(\sigma_j) & i = j \\ \min\limits_{i \le k < j} \{(m[i,k])^2 + (m[k+1,j])^2\} & \boxed{i < j} \end{cases}$$

$$m[1,n]$$

$$\boxed{s[i,j]} = \begin{cases} ((\sigma_1 \quad (\sigma_2 \quad \sigma_3)) \quad ((\sigma_4 \quad \sigma_5) \quad \sigma_6)) \end{cases}$$



$((\sigma_1 \quad (\sigma_2 \quad \sigma_3)) \quad ((\sigma_4 \quad \sigma_5) \quad \sigma_6))$