Modulo: Strutture dati dinamiche lineari [P2_08]

Unità didattica: Applicazione delle liste ad altre strutture dati [8-AT]

Titolo: Uso delle liste per la realizzazione di altre strutture dati lineari

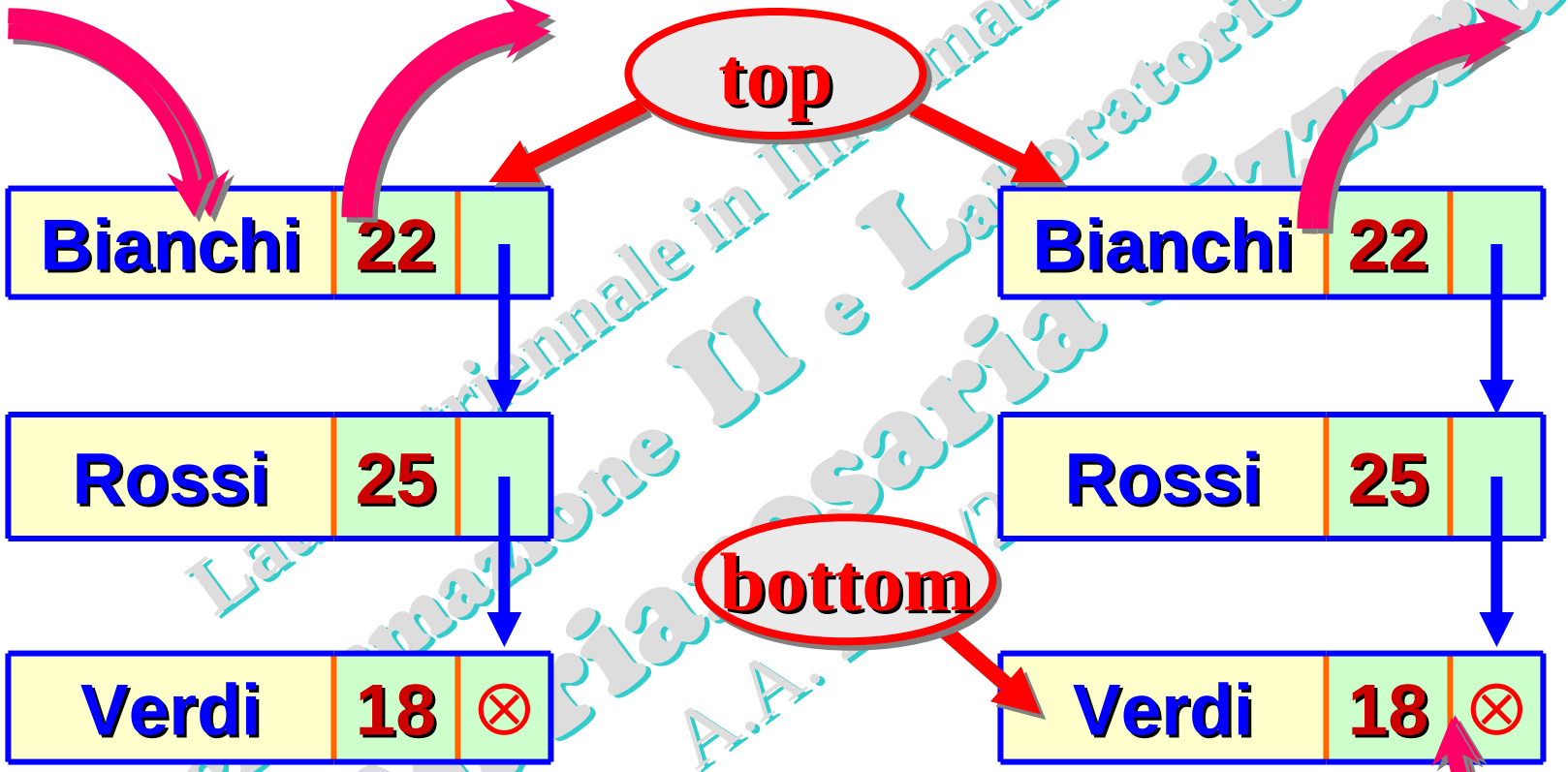
Argomenti trattati:

- ✓ Pila e coda mediante la struttura dati lista lineare
- ✓ Lista circolare e lista bidirezionale
- ✓ Lista multipla (multilista)
- ✓ Matrici sparse rappresentate come liste multiple con accesso indipendente sia per riga e sia per colonna
- ✓ Matrici sparse rappresentate come liste multiple con accesso esclusivo solo per riga oppure solo per colonna

Prerequisiti richiesti: implementazione C di una lista lineare, strutture dati dinamiche lineari, matrici

pila (o stack) (mediante lista)

coda (o queue) (mediante lista)

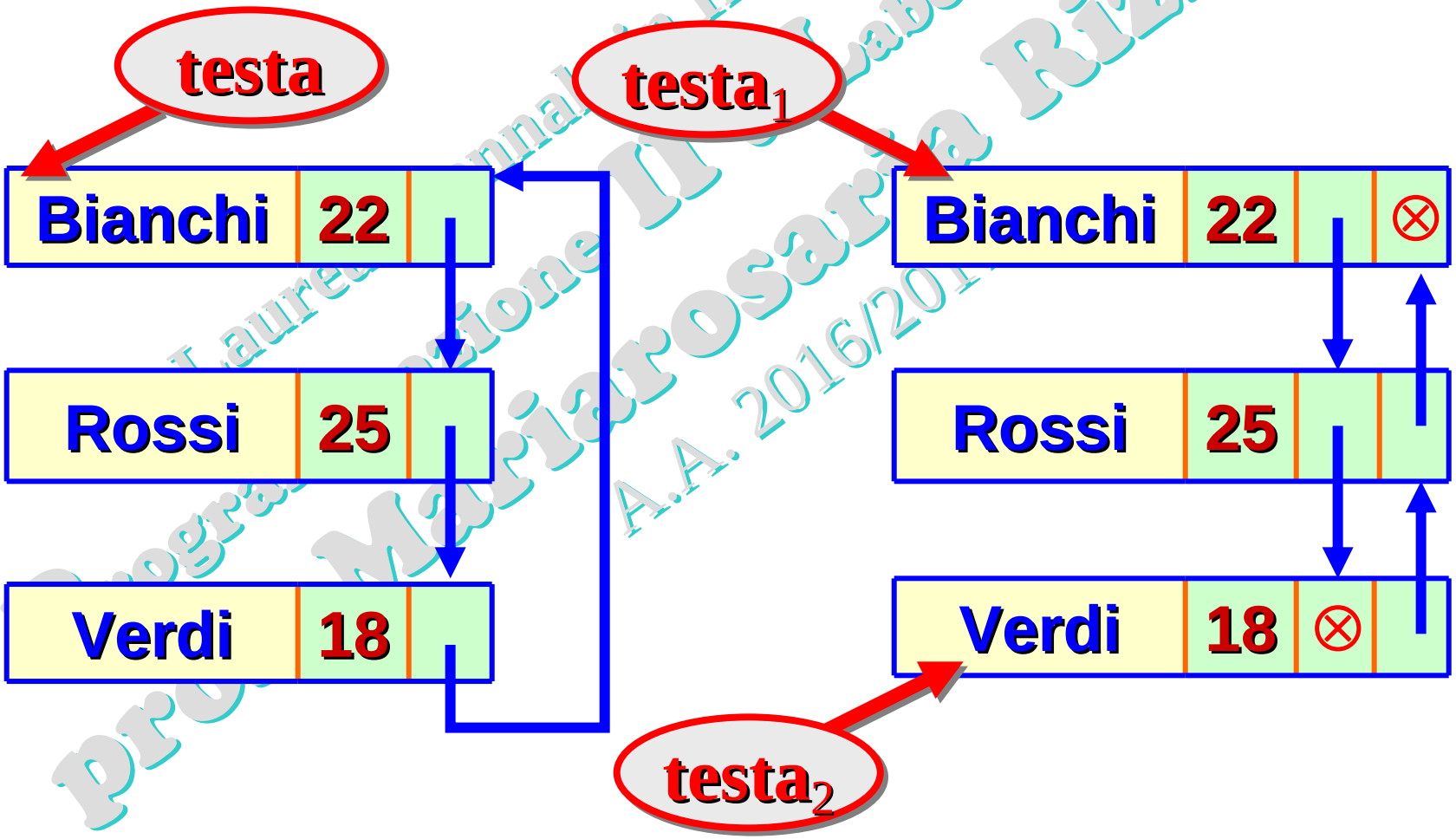


Lista con inserimento / eliminazione sempre all'inizio.

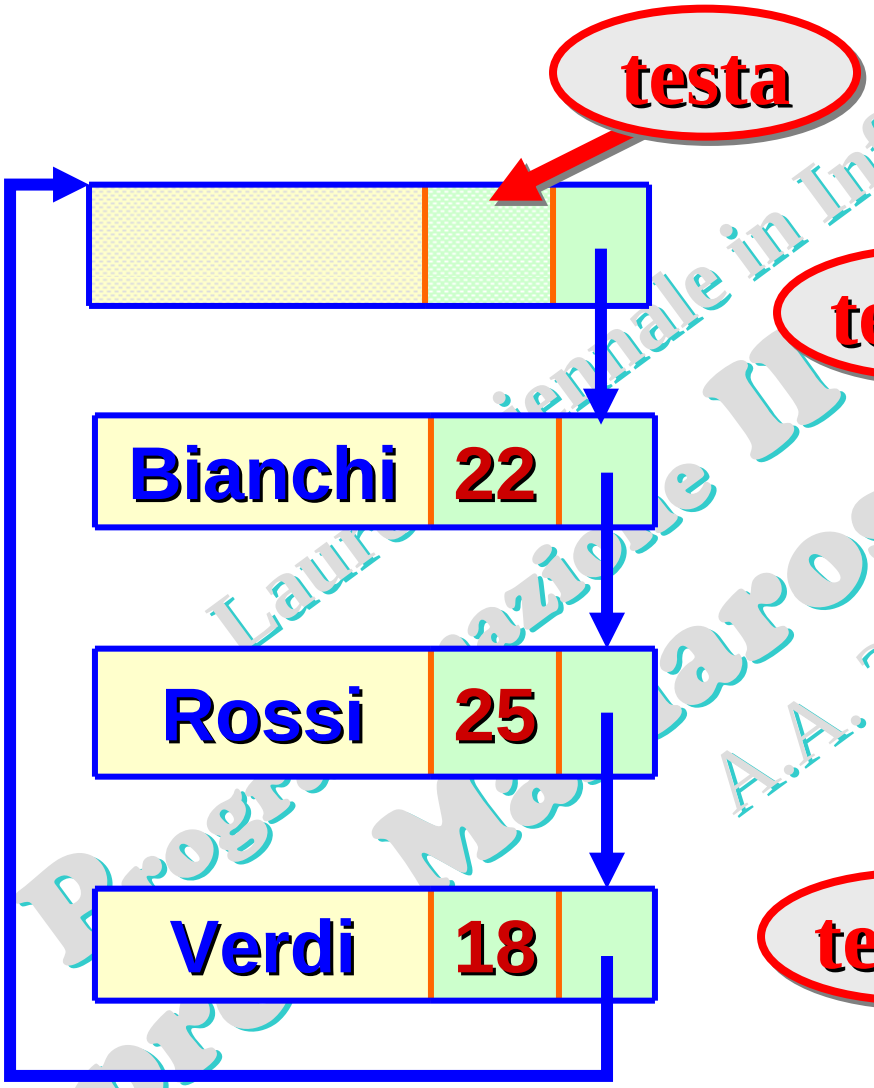
Lista con inserimento sempre all'inizio, eliminazione sempre alla fine.

lista circolare (o catena)

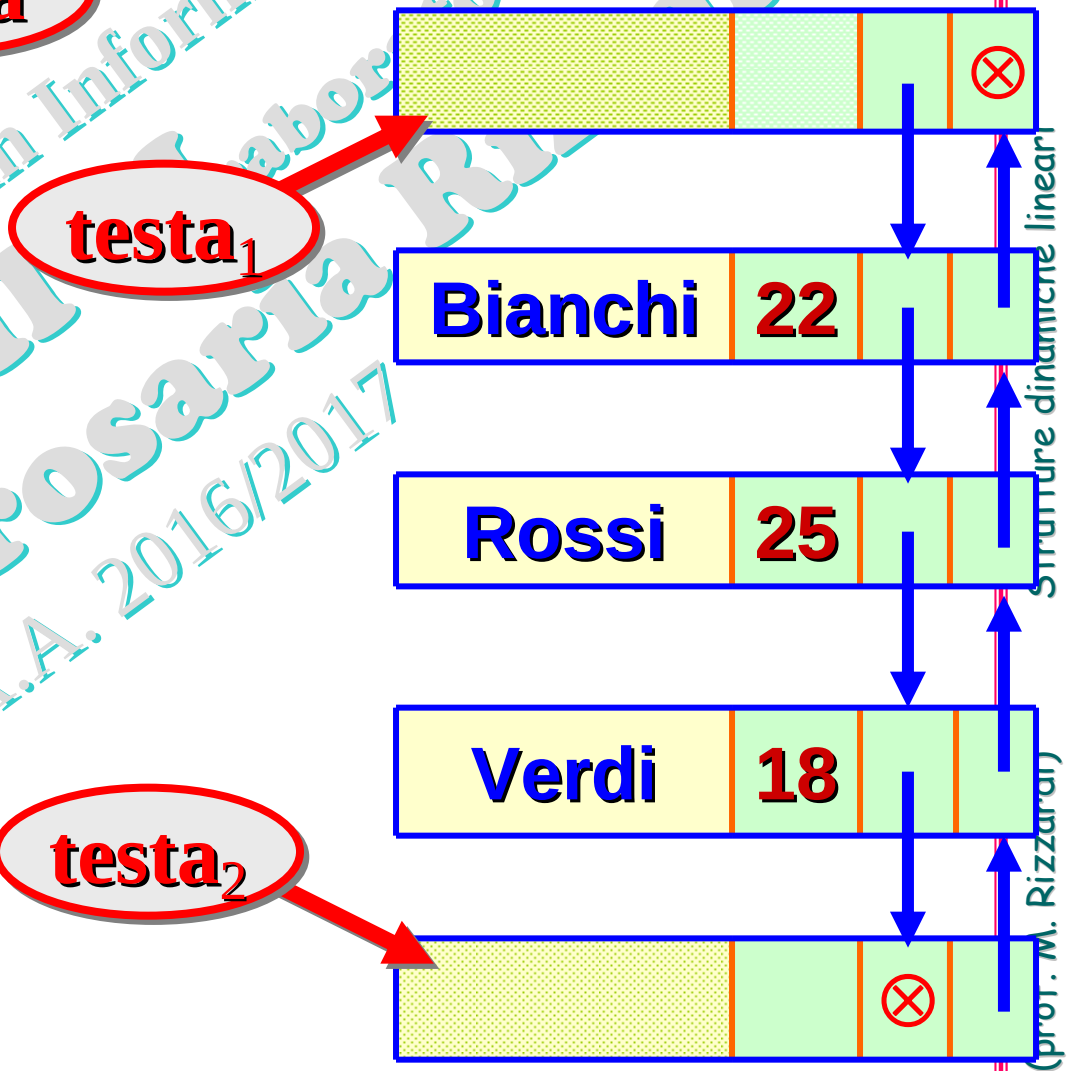
lista bidirezionale (o simmetrica)



lista circolare (o catena) con sentinella



lista bidirezionale (o doppiamente concatenata) con sentinella



Esercizio:

1

Realizzare in ***C*** le funzioni per la gestione delle precedenti strutture dati:

- pila, coda [liv. 1]
- catena, lista bidirezionale [liv. 2]

mediante *linked list* dinamica e generica con e senza nodo sentinella.

Esempio di liste multiple: rappresentazione di matrici sparse

in generale una matrice sparsa ha

…molti zeri!!!

per cui non conviene memorizzarla come array 2D

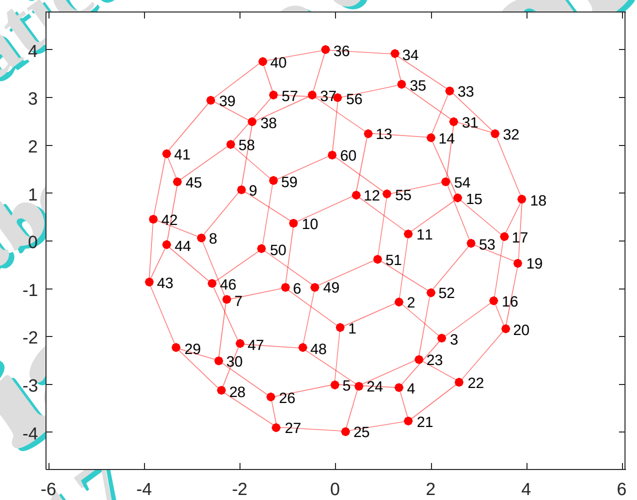
Come rappresentare una matrice sparsa in memoria (senza memorizzare gli zeri) consentendo di accedere ai suoi elementi indifferentemente per righe e per colonne?

Esempio di matrice sparsa

MATLAB

```
B = bucky; G=graph(B);
figure; plot(G,'r'); axis equal
```

B è una matrice sparsa che rappre-
senta il pallone del calcio



B è una matrice sparsa

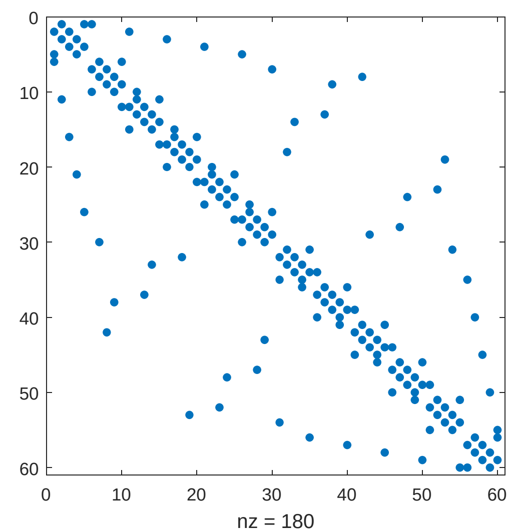
```
disp(size(B))
    60    60
disp(numel(B))
          3600
disp(nnz(B))
   180
disp(nnz(B)/numel(B))
           0.05
figure; spy(B)
```

di size 60×60

numero tot. elementi

numero elementi non nulli

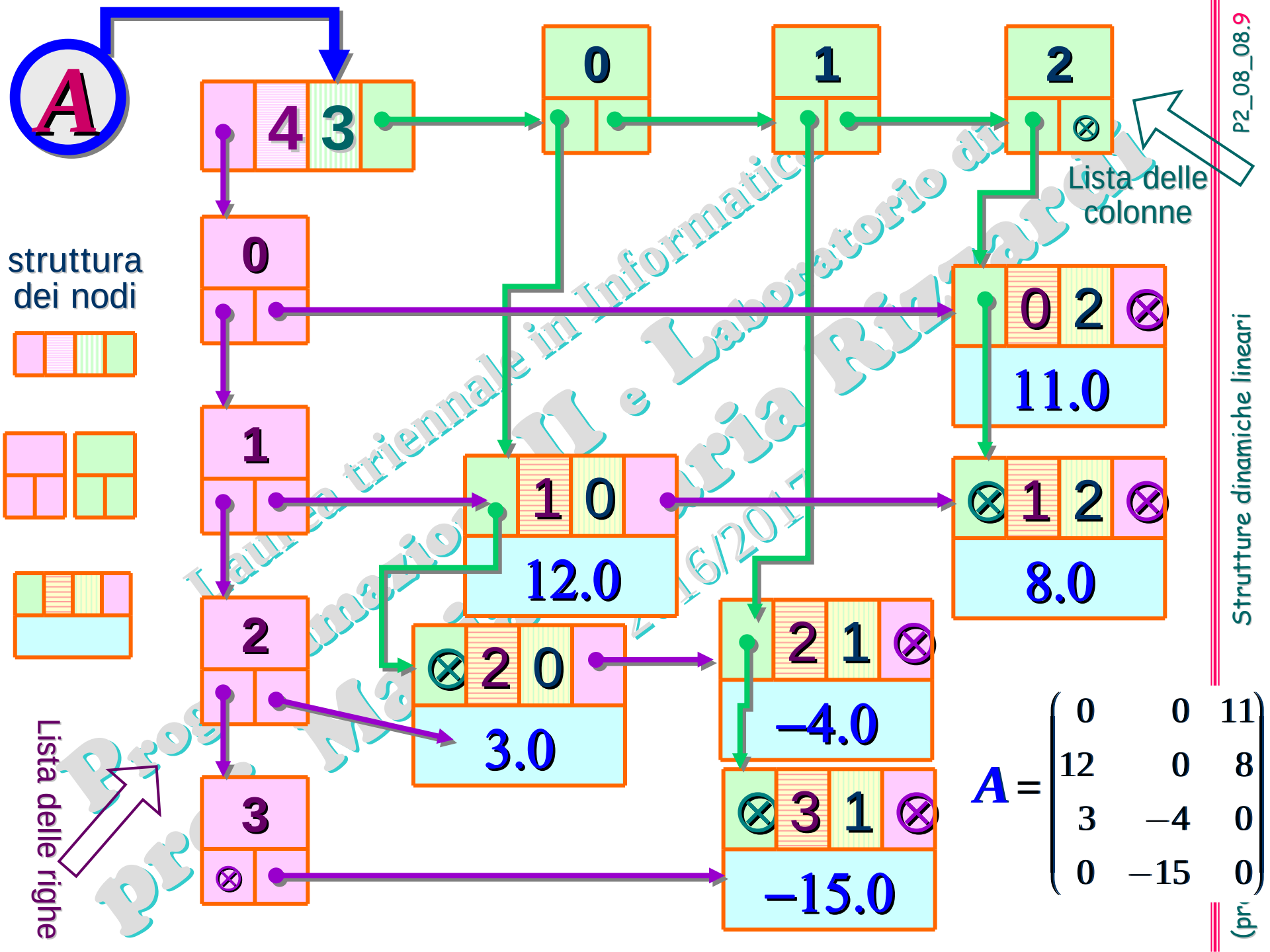
percentuale di sparsità 5%



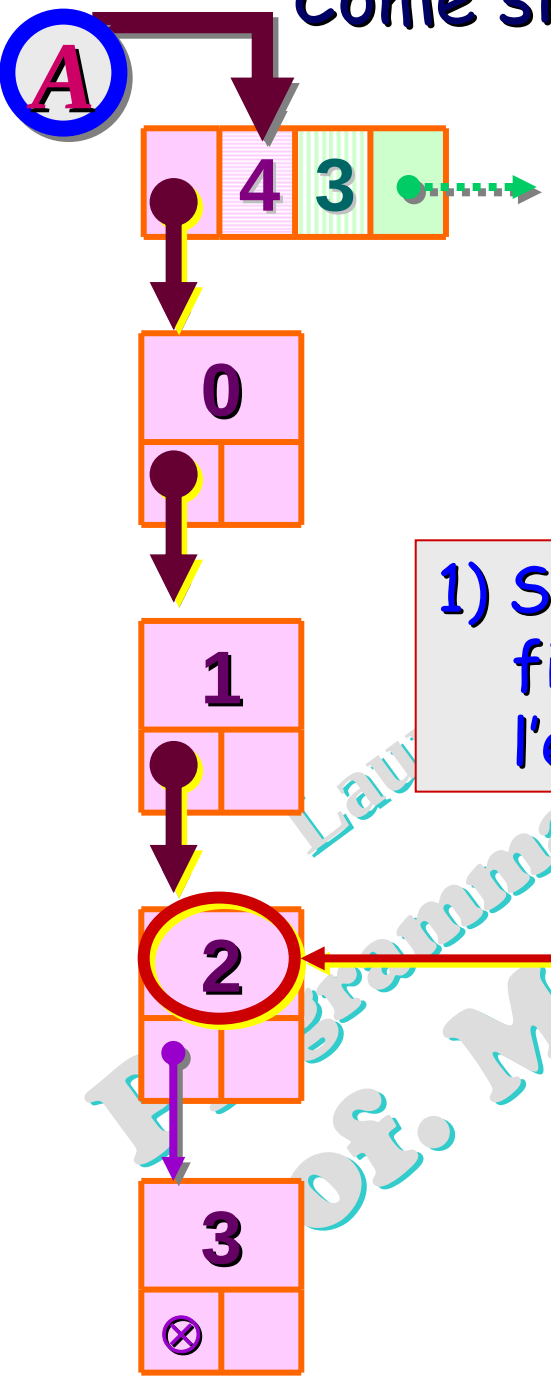
Esempio di liste multiple: rappresentazione di matrici sparse

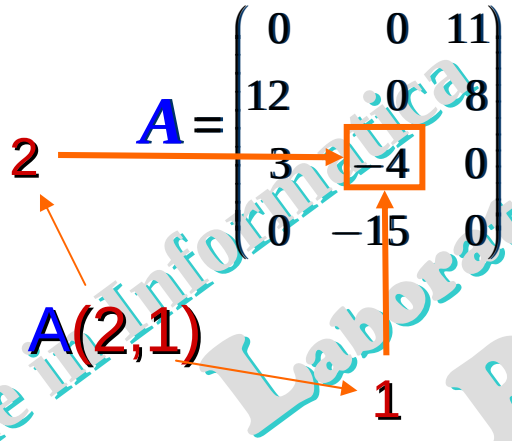
$$A = \begin{pmatrix} 0 & 0 & 11 \\ 12 & 0 & 8 \\ 3 & -4 & 0 \\ 0 & -15 & 0 \end{pmatrix}$$

Come rappresentare una matrice sparsa in memoria (senza memorizzare gli zeri) consentendo di accedere ai suoi elementi indifferentemente per righe e per colonne?


A
4 3
0
1
2
Lista delle colonne
struttura dei nodi
0
1
2
0 2
11.0
1 0
12.0
1 2
8.0
2 0
3.0
2 1
–4.0
3 1
–15.0
3
Lista delle righe
$A = \begin{pmatrix} 0 & 0 & 11 \\ 12 & 0 & 8 \\ 3 & -4 & 0 \\ 0 & -15 & 0 \end{pmatrix}$

Come si accede per righe ad un elemento?





1) Si avanza sulla lista dei puntatori alle righe fino a trovare quella cui appartiene l'elemento cercato

Come si accede per righe ad un elemento?

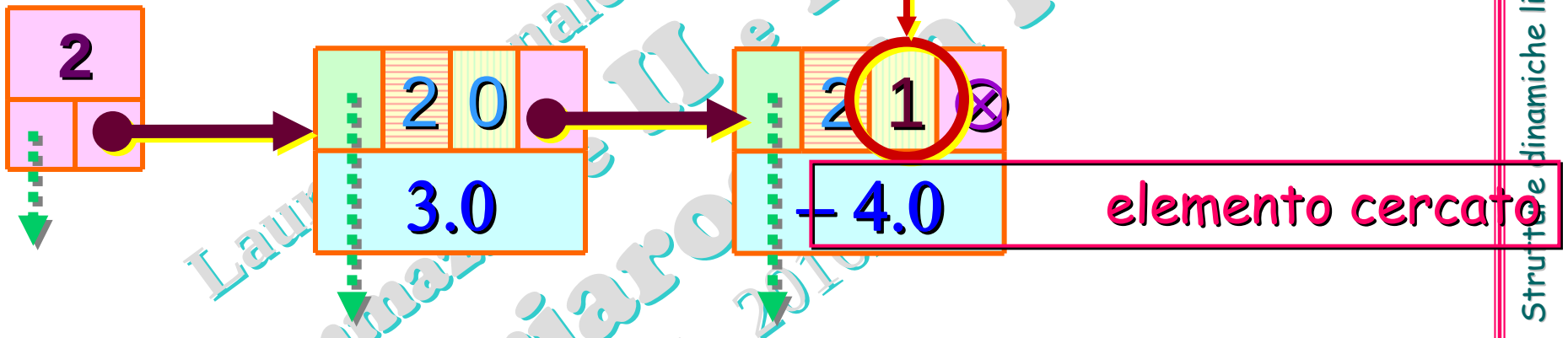
$$A = \begin{pmatrix} 0 & 0 & 11 \\ 12 & 0 & 8 \\ 3 & -4 & 0 \\ 0 & -15 & 0 \end{pmatrix}$$

2

1

A(2,1)

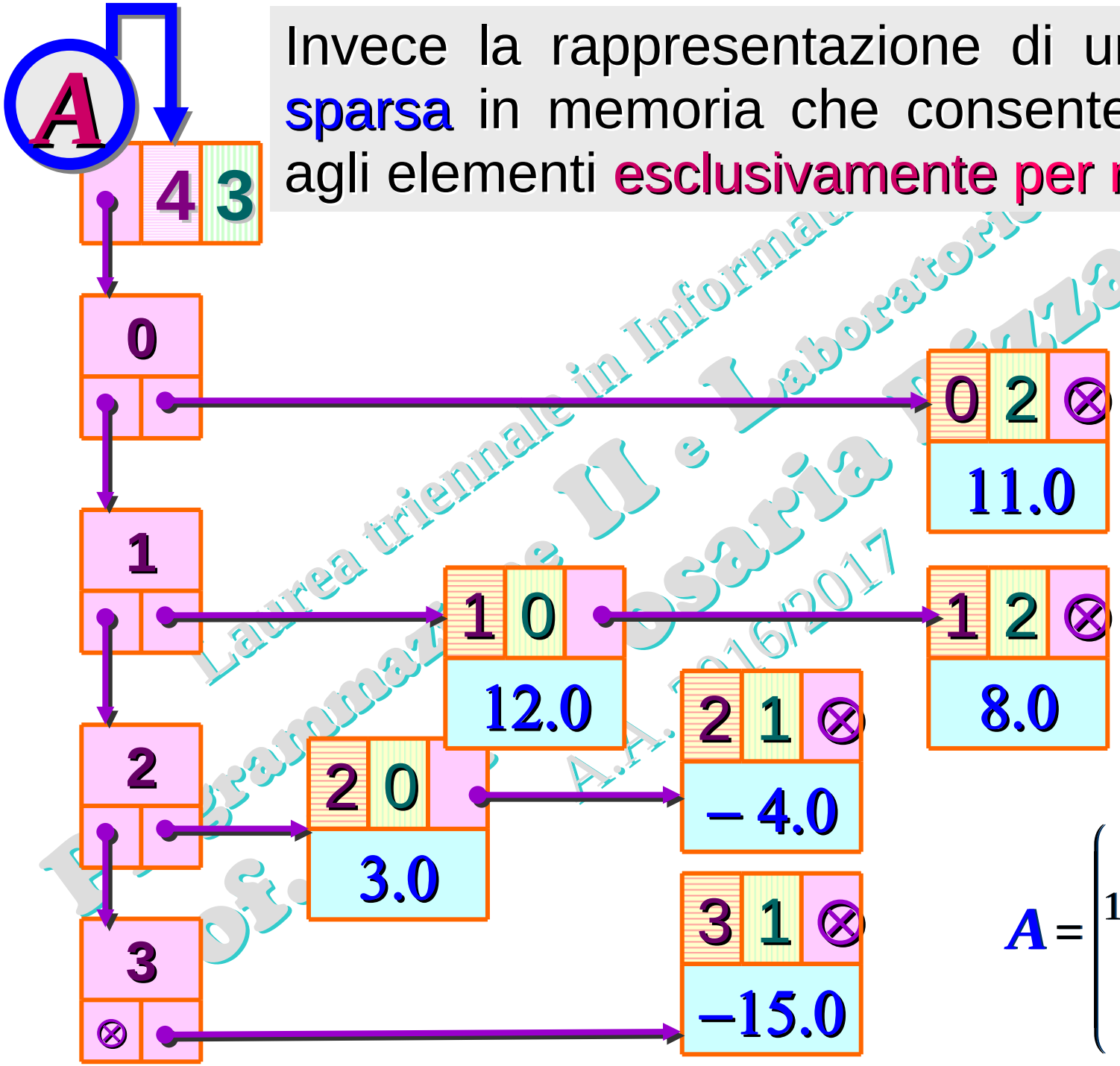
2) Si scorre la lista della riga cercando l'indice di colonna dell'elemento voluto



3) Se l'indice di colonna è presente allora si è trovato l'elemento cercato.

Se l'indice di colonna non è presente allora l'elemento cercato è zero!

Invece la rappresentazione di una **matrice sparsa** in memoria che consente l'accesso agli elementi **esclusivamente per righe** è ...



$$A = \begin{pmatrix} 0 & 0 & 11 \\ 12 & 0 & 8 \\ 3 & -4 & 0 \\ 0 & -15 & 0 \end{pmatrix}$$

Esercizio:

1

Implementare in ***C*** il prodotto righe×colonne di due matrici sparse rappresentate tramite liste multiple. Scegliere per ciascuna matrice la rappresentazione più idonea. [liv. 3]