

Unità didattica: generalità sulla struttura dati albero (tree)

[1-T]

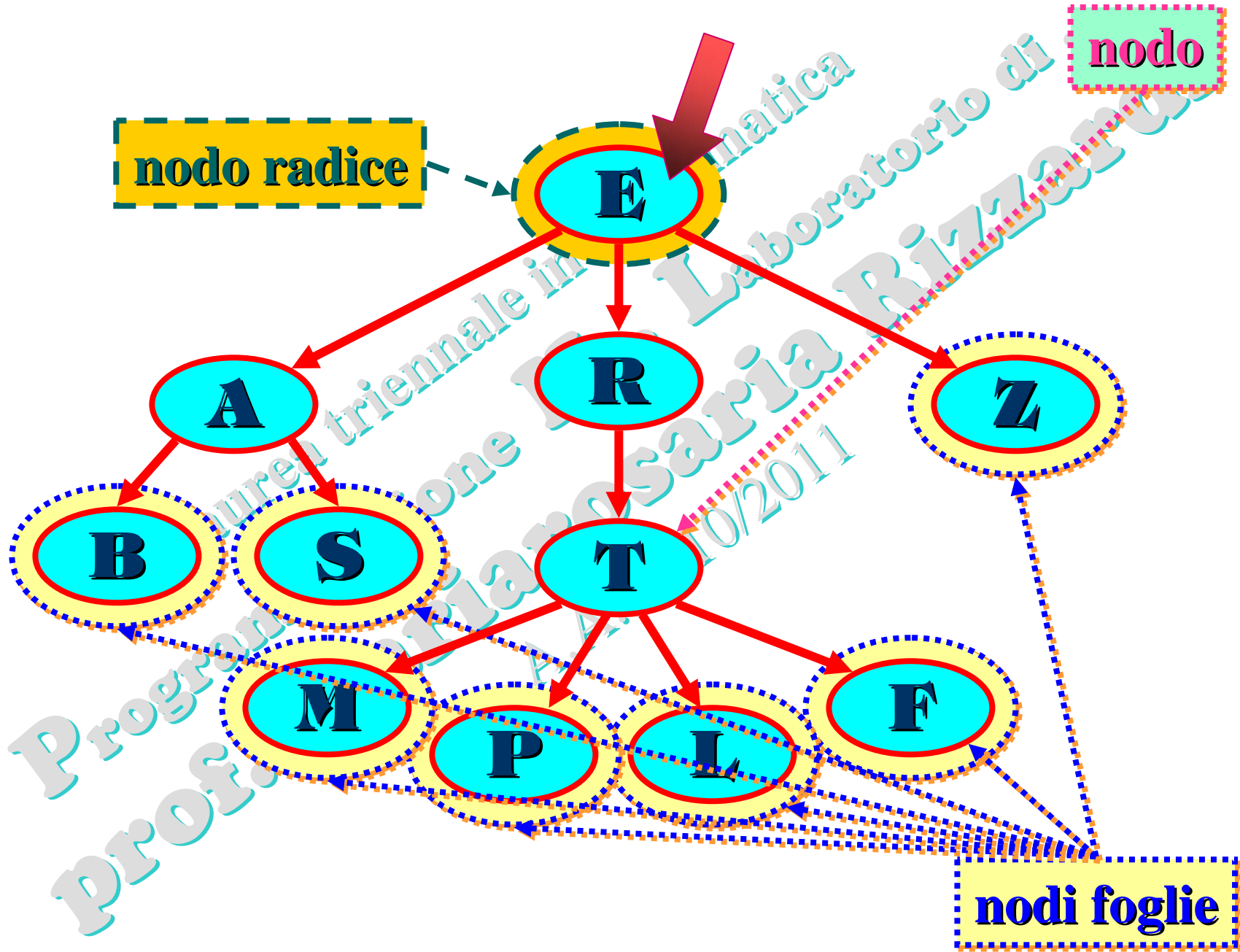
Titolo: Definizioni e proprietà

Argomenti trattati:

- ✓ Definizione di nodo radice, nodo corrente, nodo figlio, nodo padre, nodi foglie
- ✓ Grado e sottoalbero di un nodo
- ✓ Livelli di un albero ed algoritmo di visita di un albero qualsiasi per livelli

Prerequisiti richiesti: array, record, strutture dati dinamiche lineari

Struttura dati gerarchica: **albero (tree)**



nodo padre
del nodo corrente

Livello
1

Livello
2

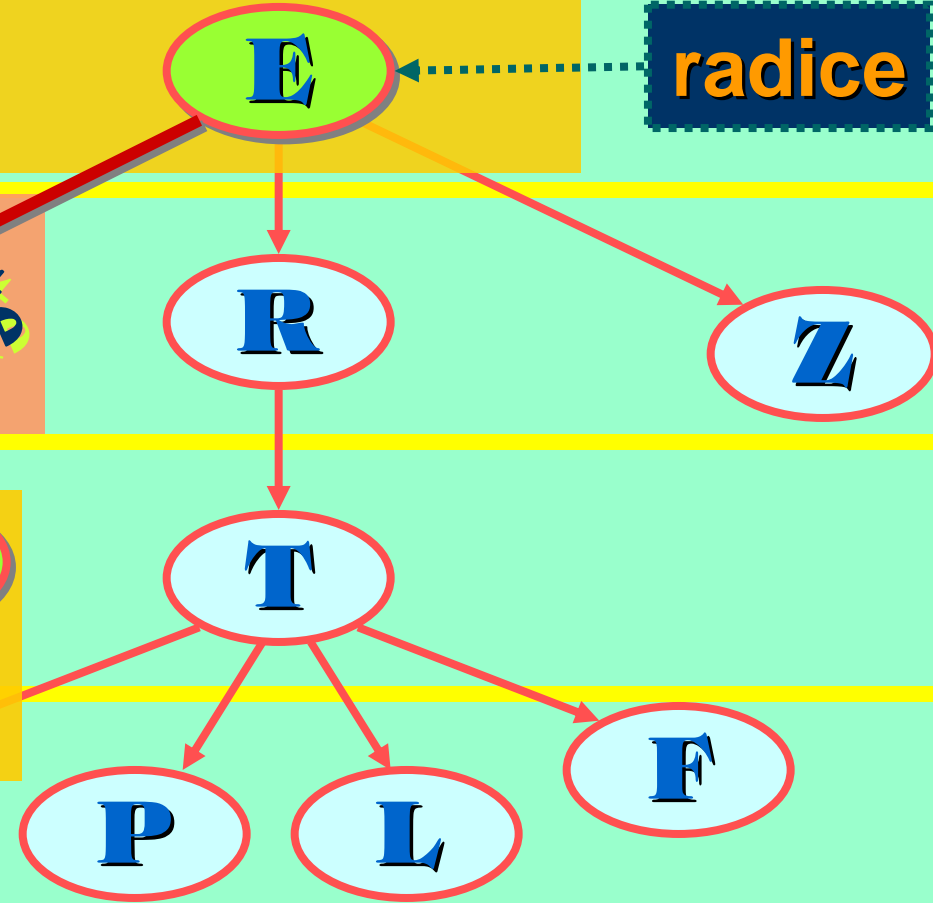
Livello
3

Livello
4

radice

nodo corrente

nodi figli
del nodo corrente

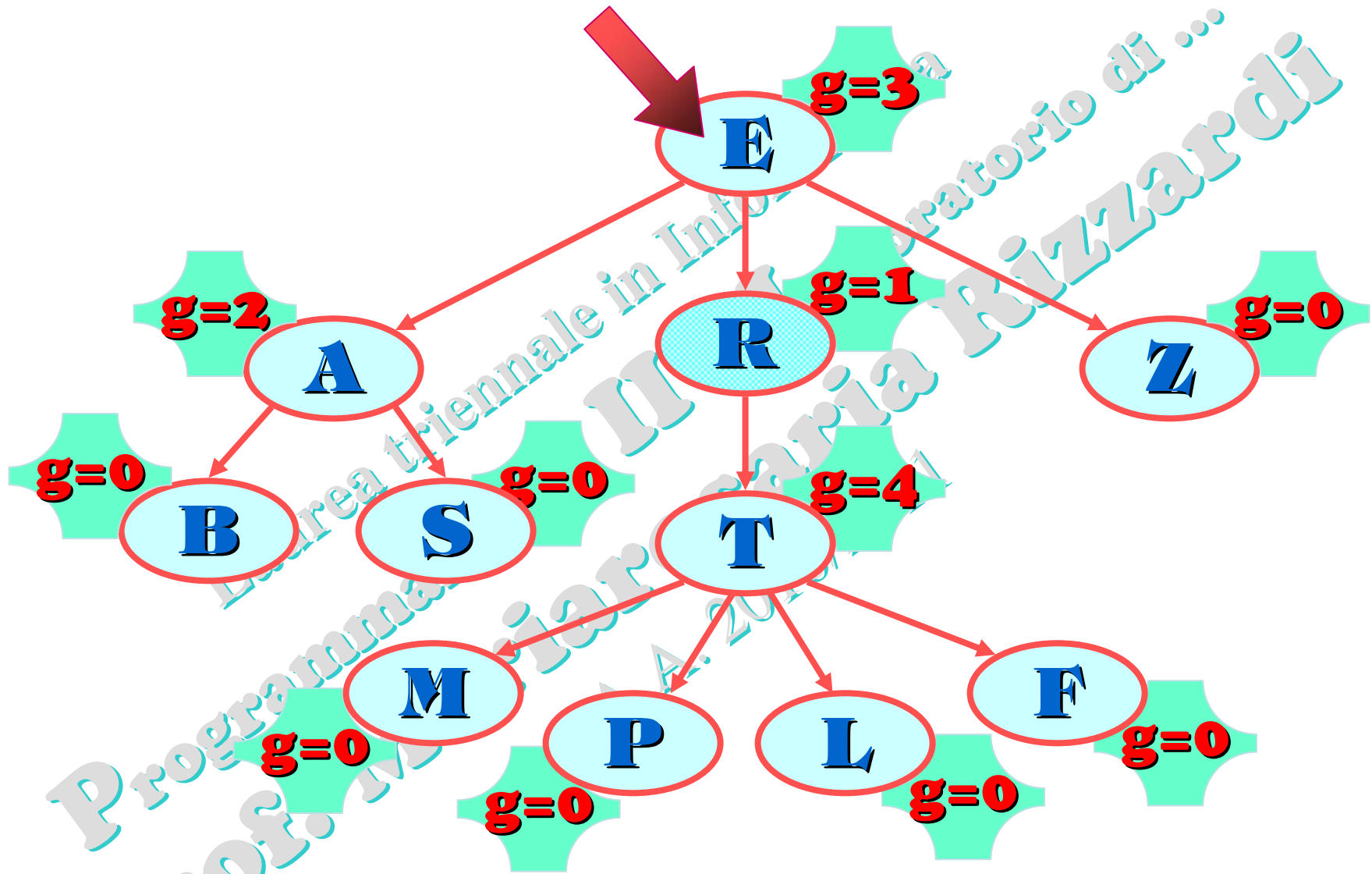


La **radice** è l'unico nodo che non ha padre.

Le **foglie** sono gli unici nodi che non hanno figli.

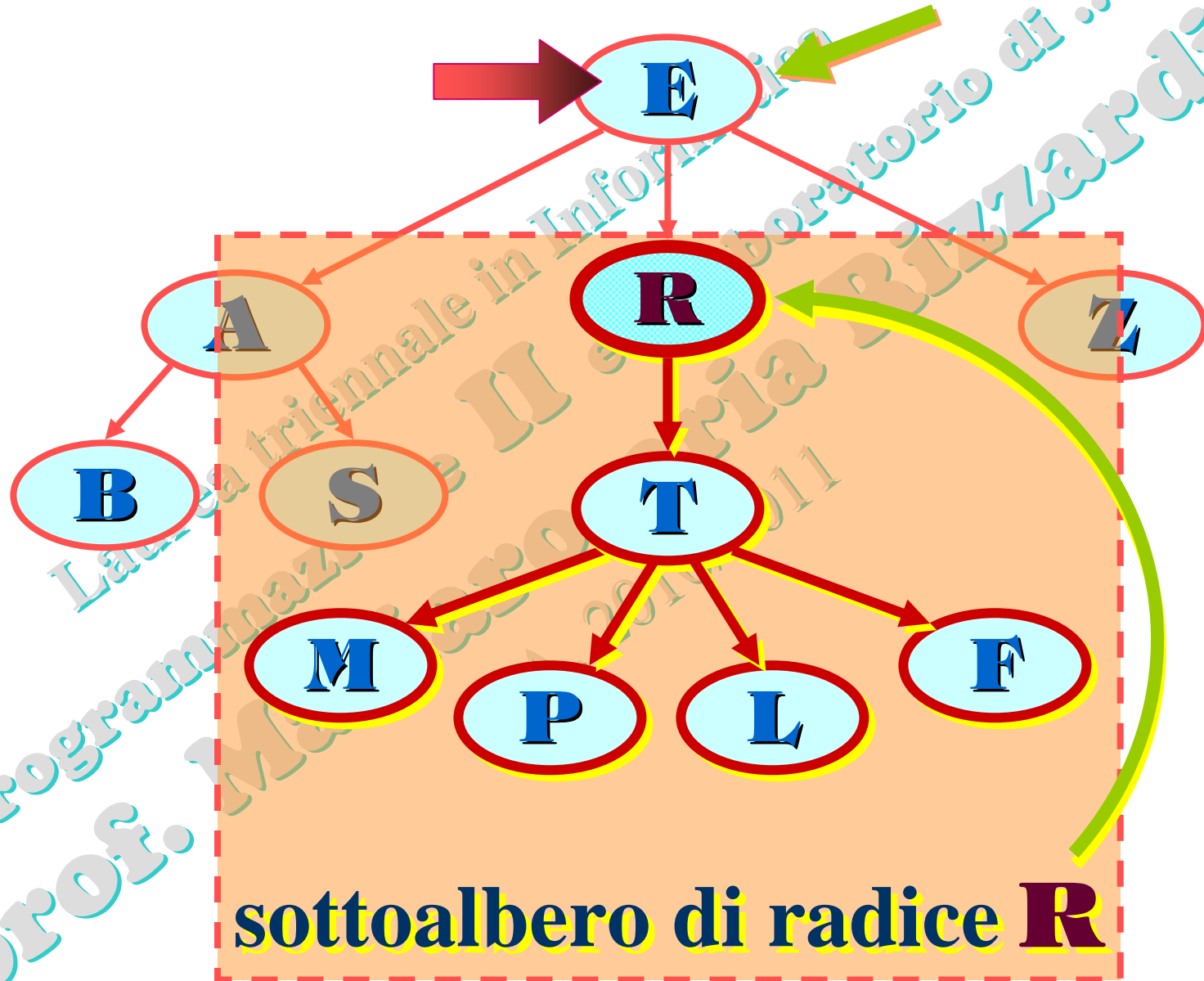
Ogni nodo (tranne la radice) ha un unico padre.

grado g di un nodo = numero di figli



le foglie hanno sempre grado 0

albero di radice **E**

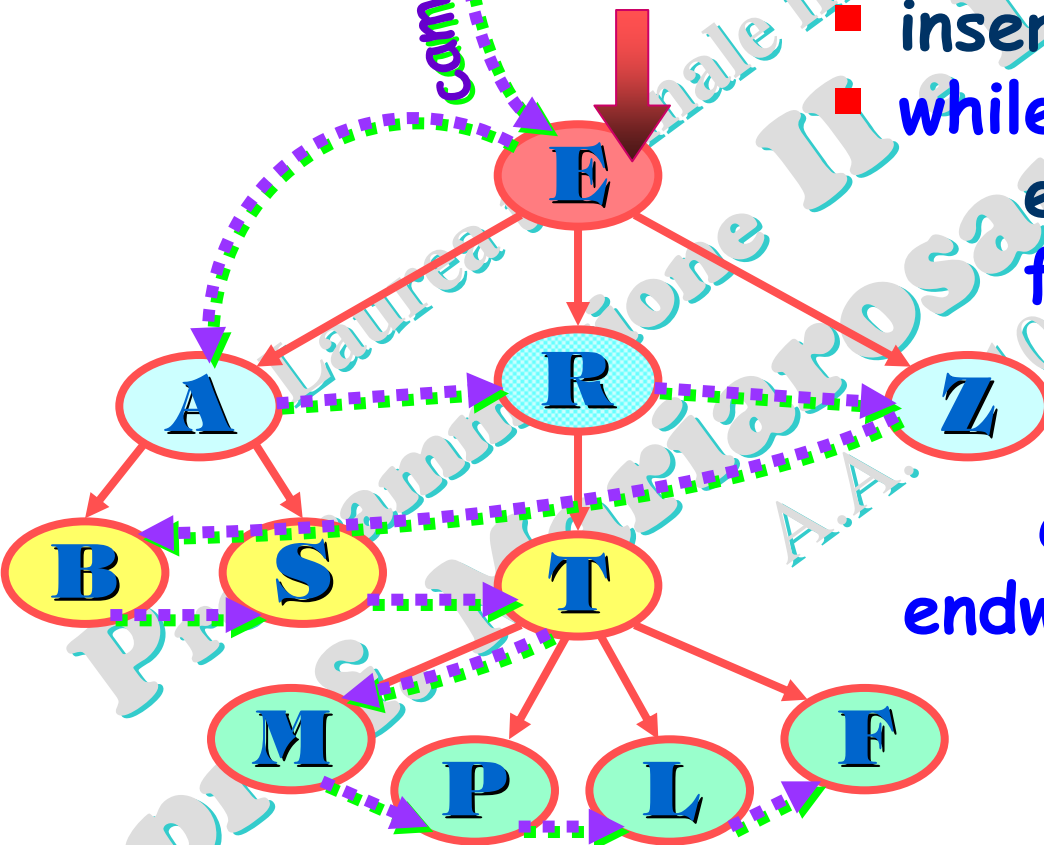


Visita di un albero (qualsiasi) per livelli

IDEA algoritmo

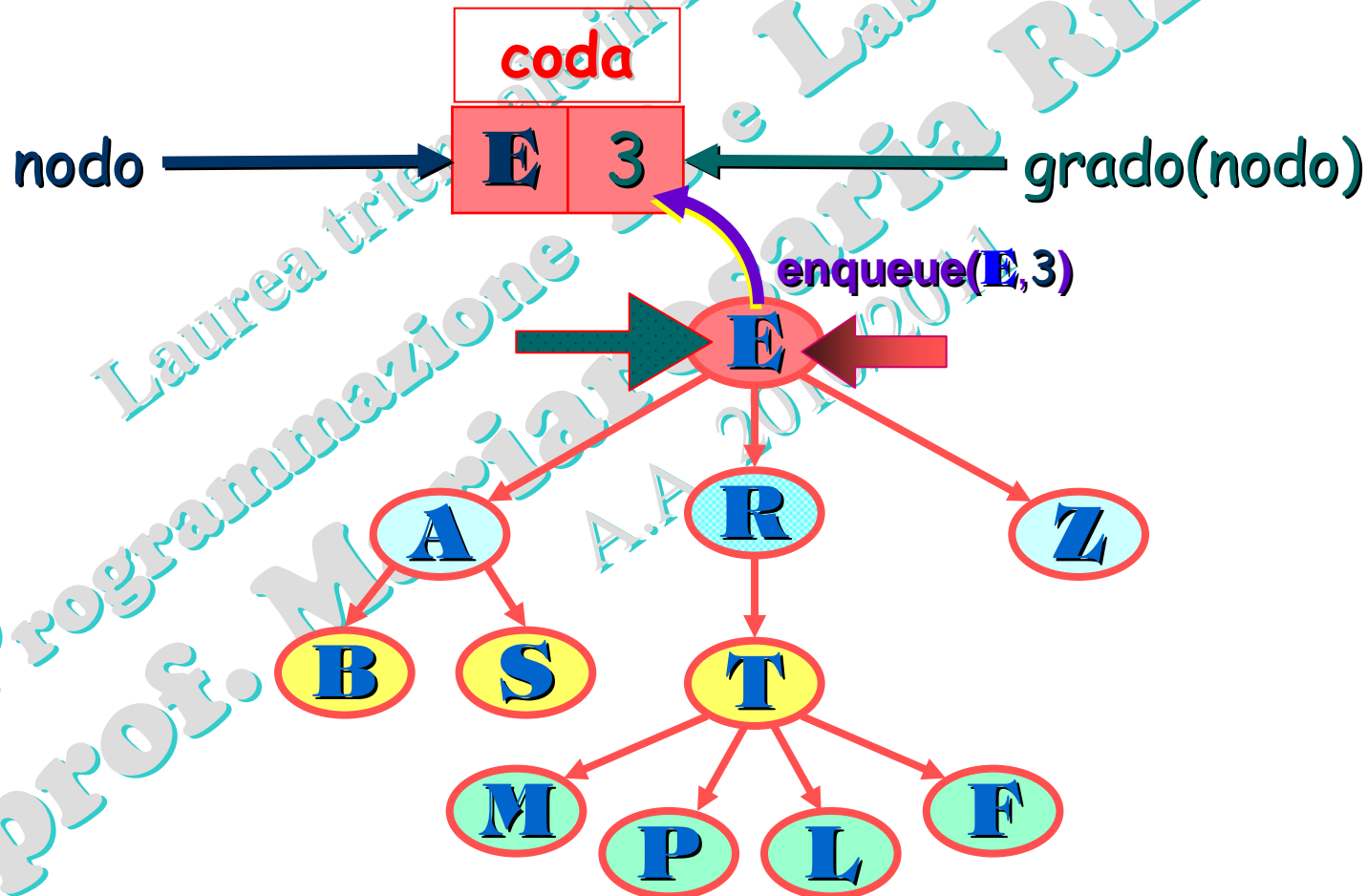
- visita la radice;
- inseriscila nella coda;
- **while** coda non vuota
 - estrai nodo dalla coda;
 - for** $k=1$ **to** grado(nodo),
 - visita figlio k^{simo} ;
 - inseriscilo nella coda;
 - endfor**
- endwhile**

cammino seguito

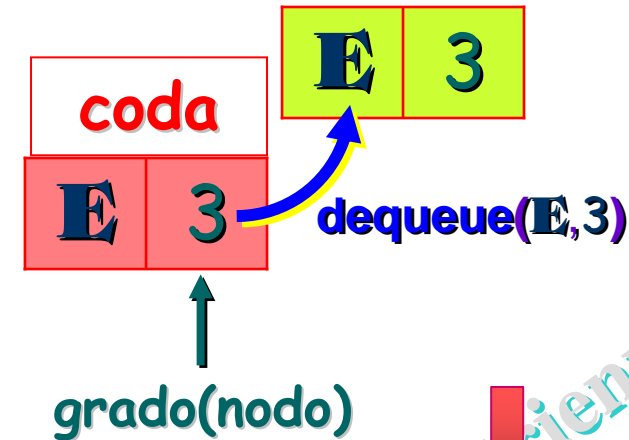


Visita di un albero per livelli [1]

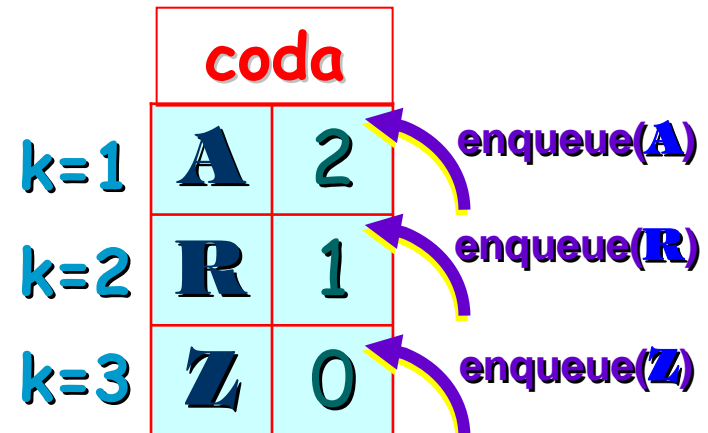
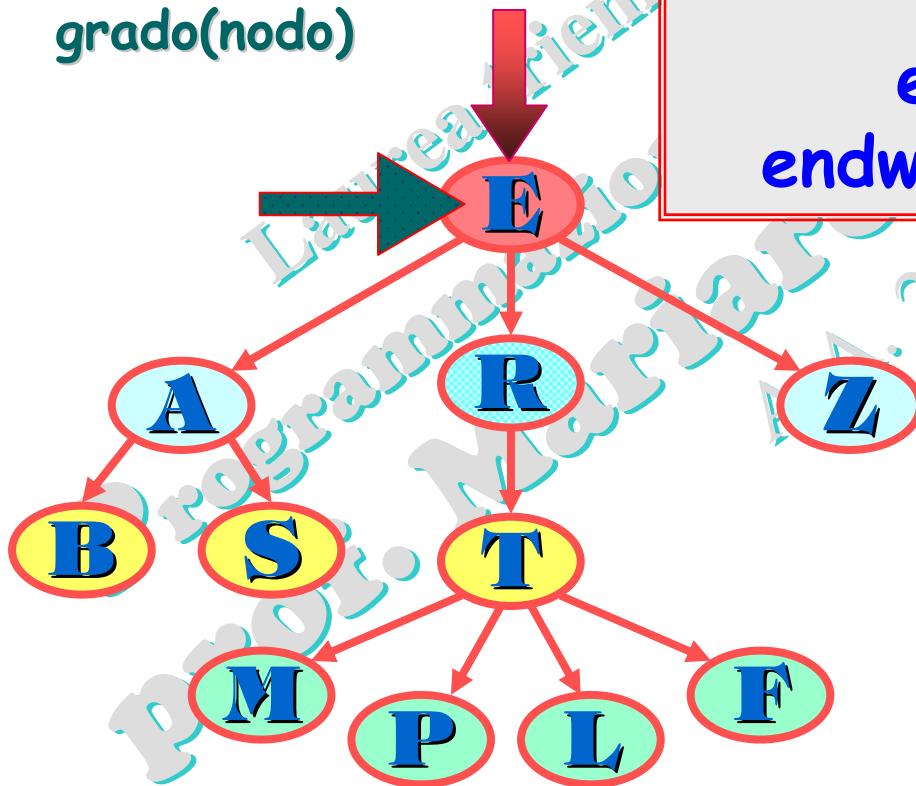
- visita la radice;
- inseriscila nella coda;



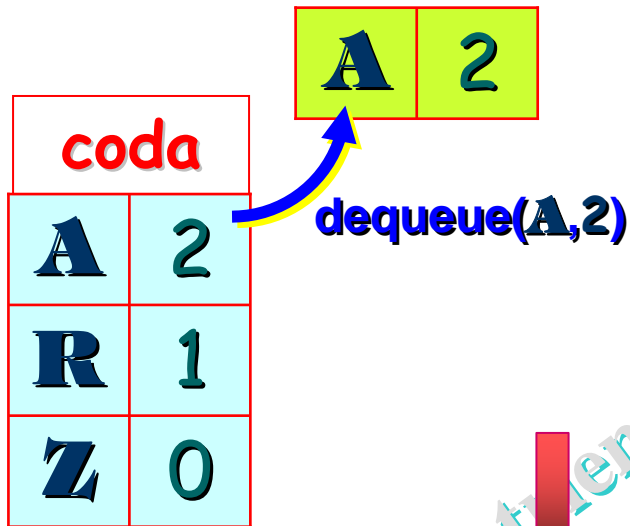
Visita di un albero per livelli [2]



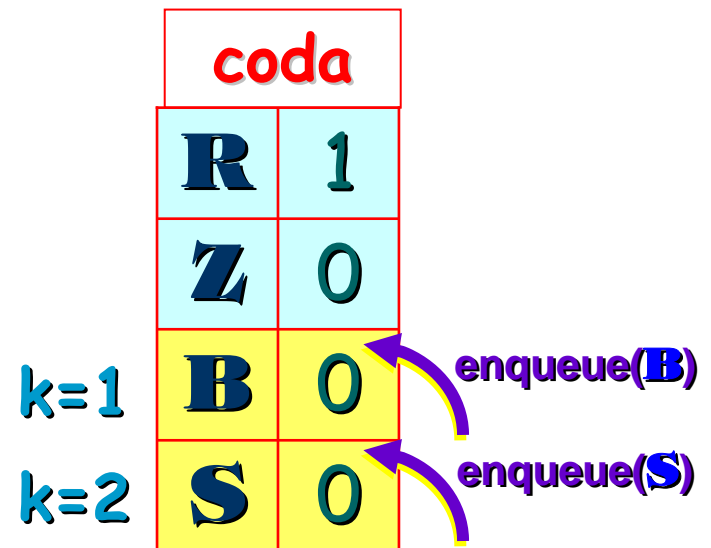
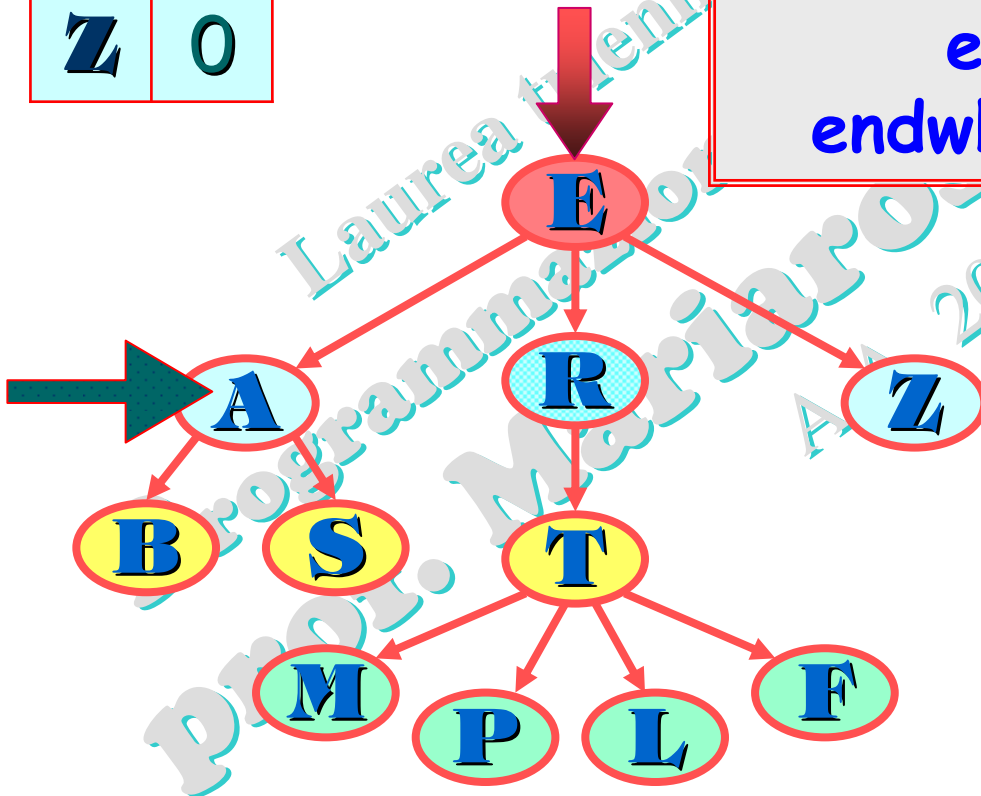
```
while coda non vuota  
  estrai nodo dalla coda;  
  for k=1 to grado(nodo),  
    visita figlio ksimo;  
    inseriscilo nella coda;  
  endfor  
endwhile
```



Visita di un albero per livelli [3]



```
while coda non vuota
  estrai nodo dalla coda;
  for k=1 to grado(nodo),
    visita figlio ksimo;
    inseriscilo nella coda;
  endfor
endwhile
```



Visita di un albero per livelli [4]

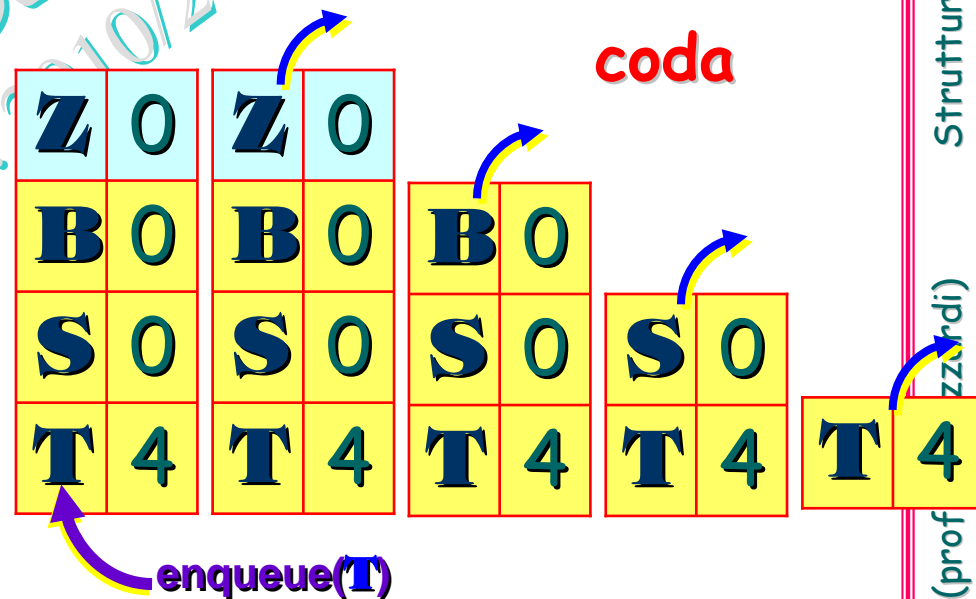
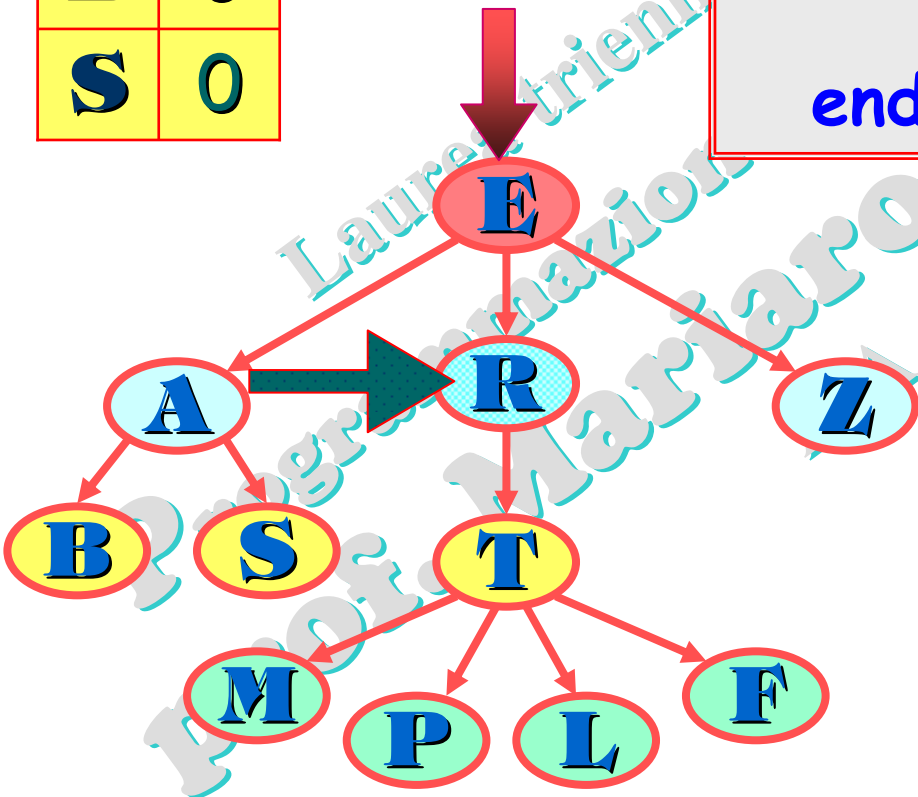
| coda | |
|----------|----------|
| R | 1 |
| Z | 0 |
| B | 0 |
| S | 0 |

| | |
|----------|----------|
| R | 1 |
|----------|----------|

dequeue(**R**,1)

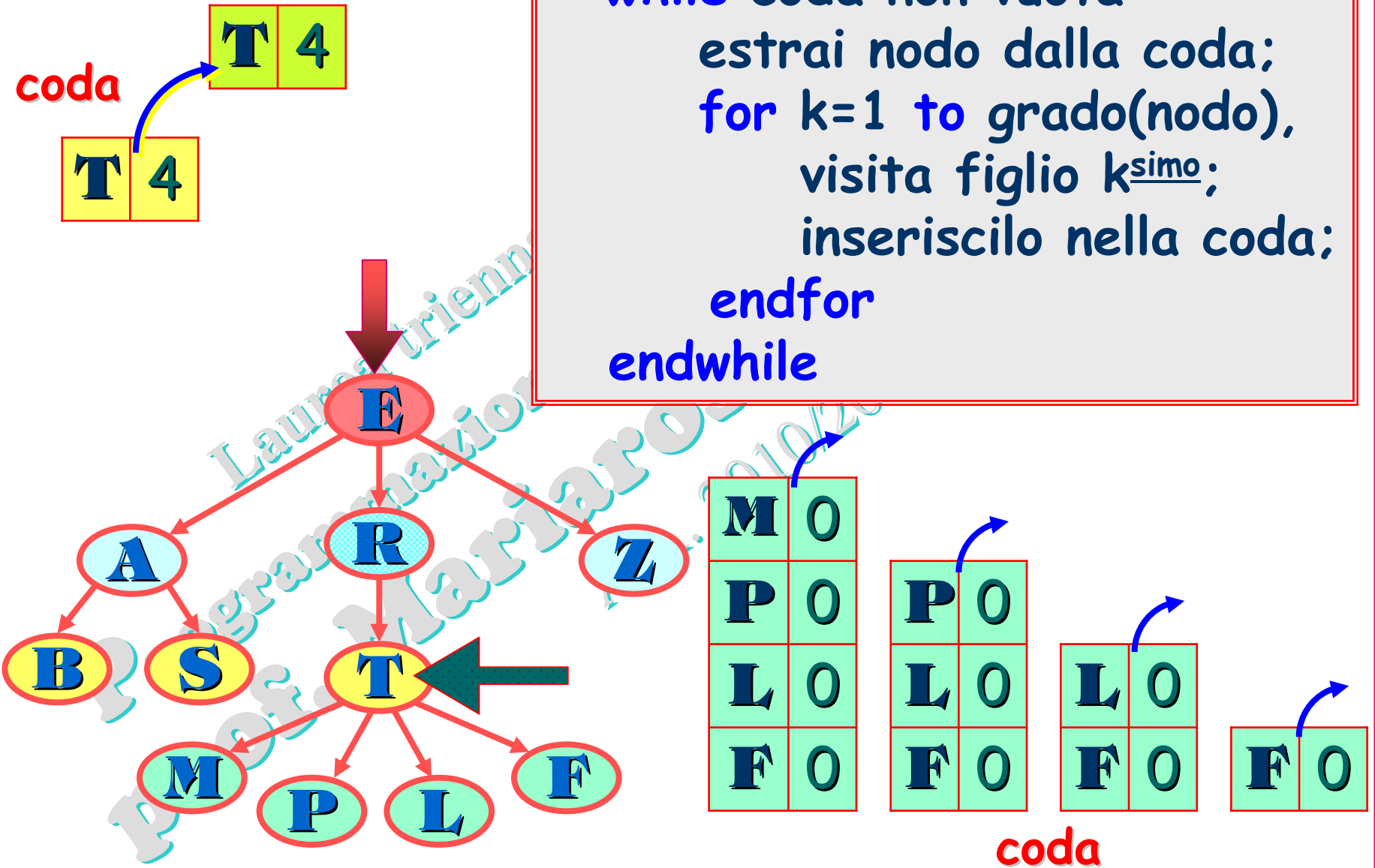
```

while coda non vuota
  estrai nodo dalla coda;
  for k=1 to grado(nodo),
    visita figlio ksimo;
    inseriscilo nella coda;
  endfor
endwhile
    
```



Visita di un albero per livelli [5]

```
■ while coda non vuota
    estrai nodo dalla coda;
    for k=1 to grado(nodo),
        visita figlio ksimo;
        inseriscilo nella coda;
    endfor
endwhile
```

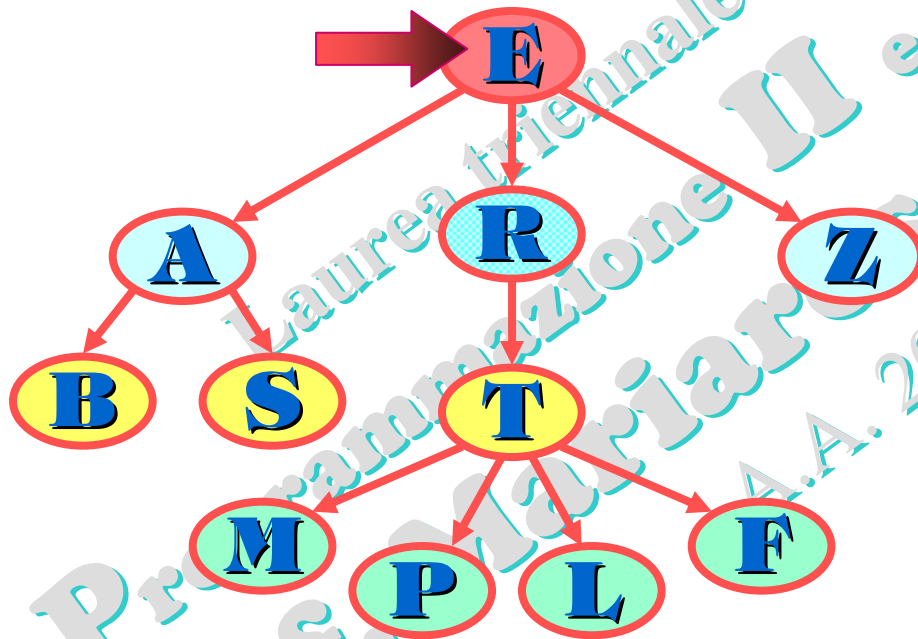


In C l'algoritmo della visita per livelli può essere utilizzato anche per la costruzione dell'albero

P2_09_01.12

INPUT: i nodi, ordinati per livello, ed il relativo grado.

OUTPUT: array di struct contenente il campo informazione di un nodo ed il suo grado.



| | | | |
|----|----------|----------|---------------|
| 0 | E | 3 | inizio |
| 1 | A | 2 | |
| 2 | R | 1 | |
| 3 | Z | 0 | |
| 4 | B | 0 | |
| 5 | S | 0 | |
| 6 | T | 4 | |
| 7 | M | 0 | |
| 8 | P | 0 | |
| 9 | L | 0 | |
| 10 | F | 0 | fine |

$g(E)=3$

$g(A)=2$

L'array può essere gestito come una coda per visitare l'albero mediante i due puntatori **inizio** e **fine**

Strutture dinamiche gerarchiche

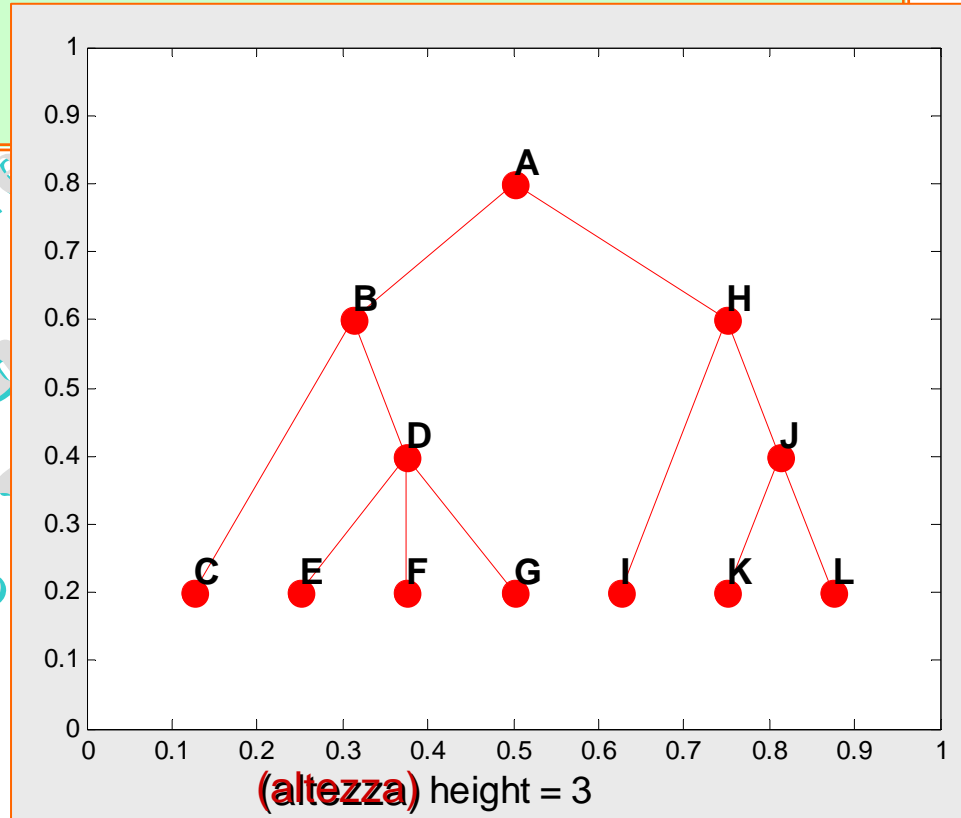
(prof. M. Rizzardi)

Esempio

In **MATLAB** la funzione **treeplot()** disegna un albero qualsiasi. L'albero è rappresentato come array dove per ogni nodo è assegnato l'indice del padre.

```
padre = [0 1 2 2 4 4 4 1 8 8 10 10];  
nodi = char(64+(1:length(padre)))'; % codici ASCII  
treeplot(padre)
```

...



| | nodo | ...di padre |
|----|------|-------------|
| 1 | A | 0 |
| 2 | B | 1 |
| 3 | C | 2 |
| 4 | D | 2 |
| 5 | E | 4 |
| 6 | F | 4 |
| 7 | G | 4 |
| 8 | H | 1 |
| 9 | I | 8 |
| 10 | J | 8 |
| 11 | K | 10 |
| 12 | L | 10 |

Esercizio:

1

Scrivere *function C* per la costruzione e la visita per livelli di un albero qualsiasi (rappresentato mediante array).

In input si conosce per ogni nodo l'indice del padre. [liv. 2]

[**suggerimento:** la struct che definisce il generico nodo dell'albero deve contenere i seguenti campi: l'informazione, il suo grado ed un array di puntatori di dimensione pari al massimo grado dei nodi]



help

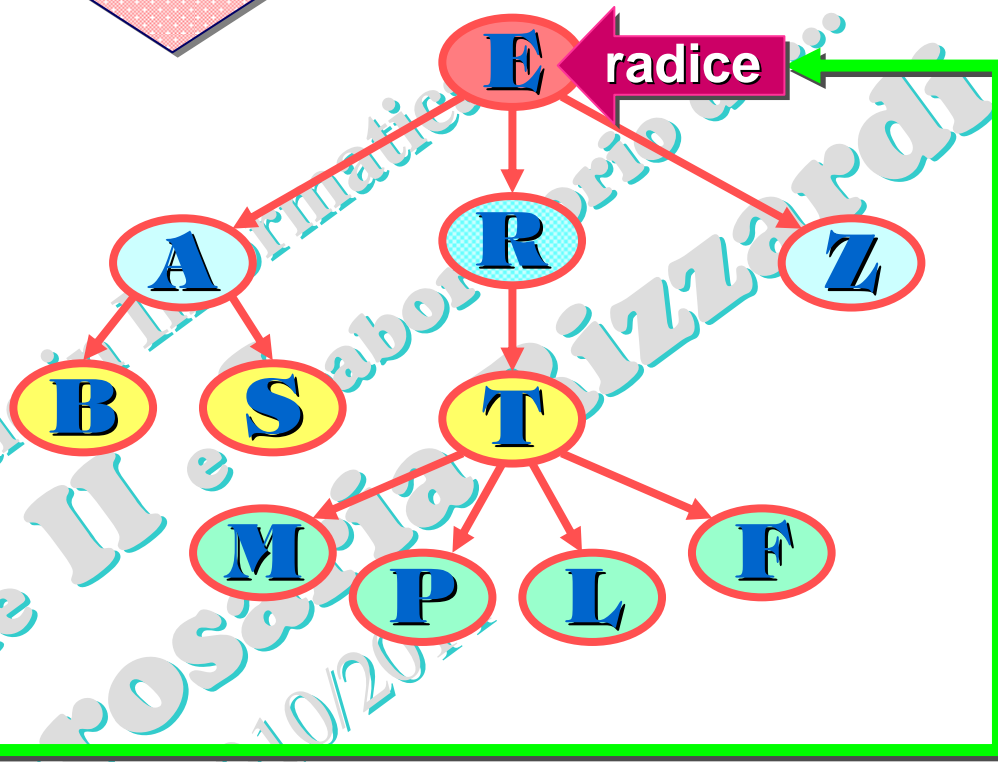
Dati di input:
 nodi in ordine qualsiasi

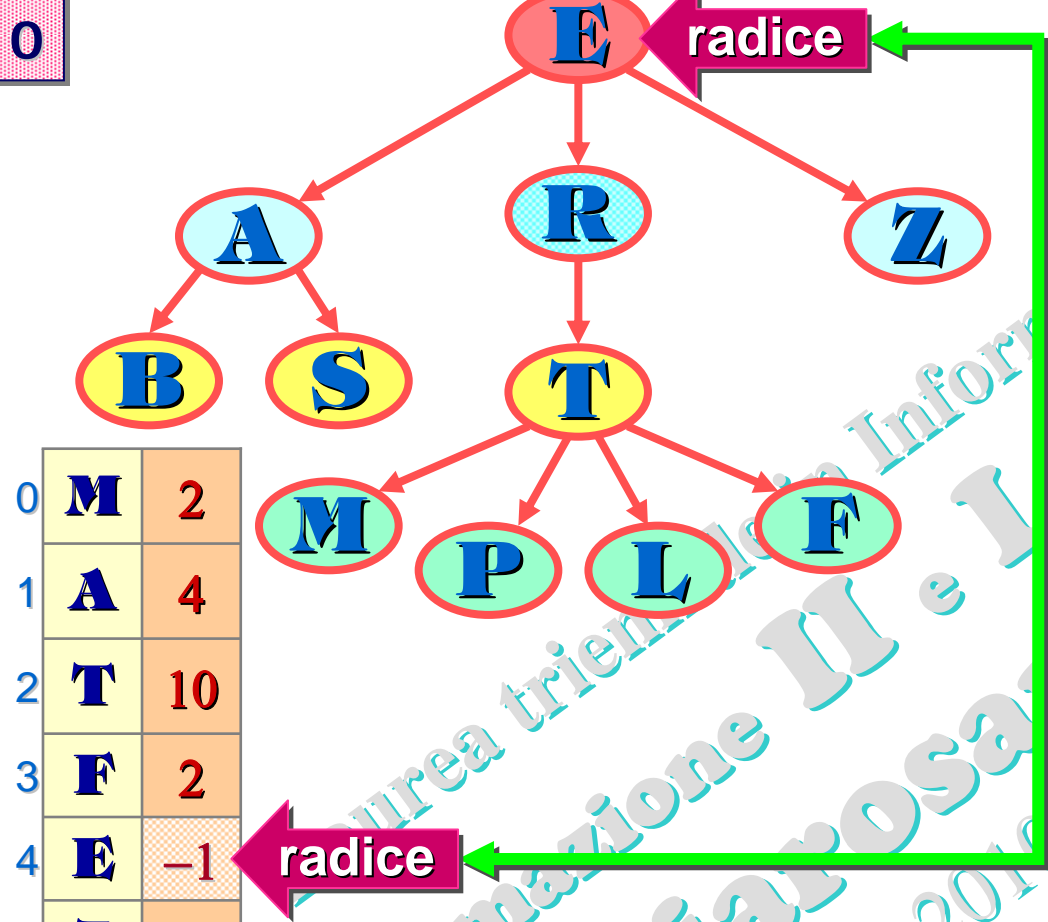
- informazione del nodo,
- padre del nodo

help

| | | |
|----|----------|----|
| 0 | M | 2 |
| 1 | A | 4 |
| 2 | T | 10 |
| 3 | F | 2 |
| 4 | E | -1 |
| 5 | Z | 4 |
| 6 | L | 2 |
| 7 | S | 1 |
| 8 | B | 1 |
| 9 | P | 2 |
| 10 | R | 4 |

radice





| | | |
|----|---|----|
| 0 | M | 2 |
| 1 | A | 4 |
| 2 | T | 10 |
| 3 | F | 2 |
| 4 | E | -1 |
| 5 | Z | 4 |
| 6 | L | 2 |
| 7 | S | 1 |
| 8 | B | 1 |
| 9 | P | 2 |
| 10 | R | 4 |

radice

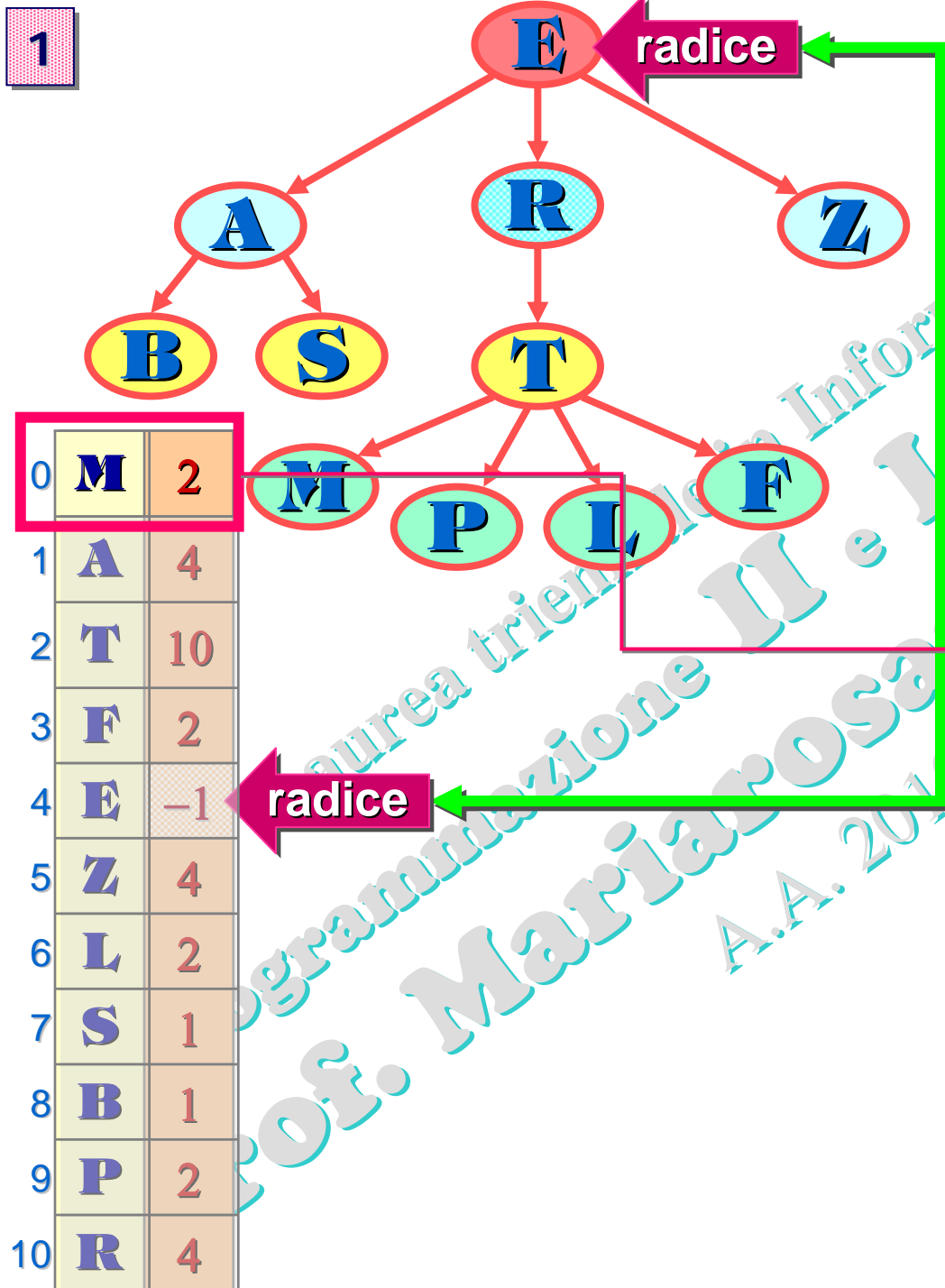
situazione iniziale
tabella

nodo

informazione

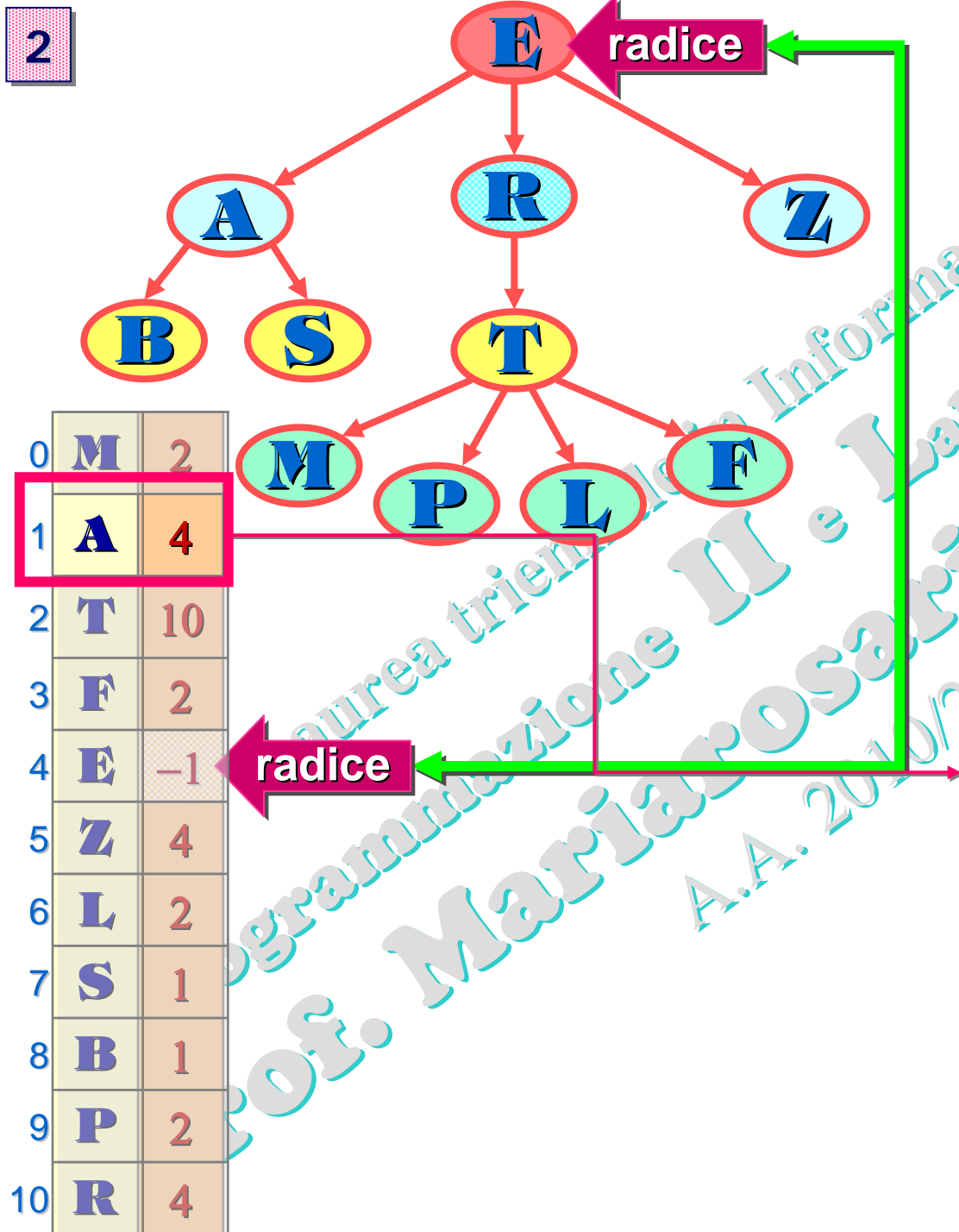
| | | | | |
|-------|-----------------|-----------------|-----------------|-----------------|
| grado | pt ₁ | pt ₂ | pt ₃ | pt ₄ |
|-------|-----------------|-----------------|-----------------|-----------------|

| INF | GR | pt ₁ | pt ₂ | pt ₃ | pt ₄ |
|-----|----|-----------------|-----------------|-----------------|-----------------|
| 0 | 0 | | | | |
| 1 | 0 | | | | |
| 2 | 0 | | | | |
| 3 | 0 | | | | |
| 4 | 0 | | | | |
| 5 | 0 | | | | |
| 6 | 0 | | | | |
| 7 | 0 | | | | |
| 8 | 0 | | | | |
| 9 | 0 | | | | |
| 10 | 0 | | | | |



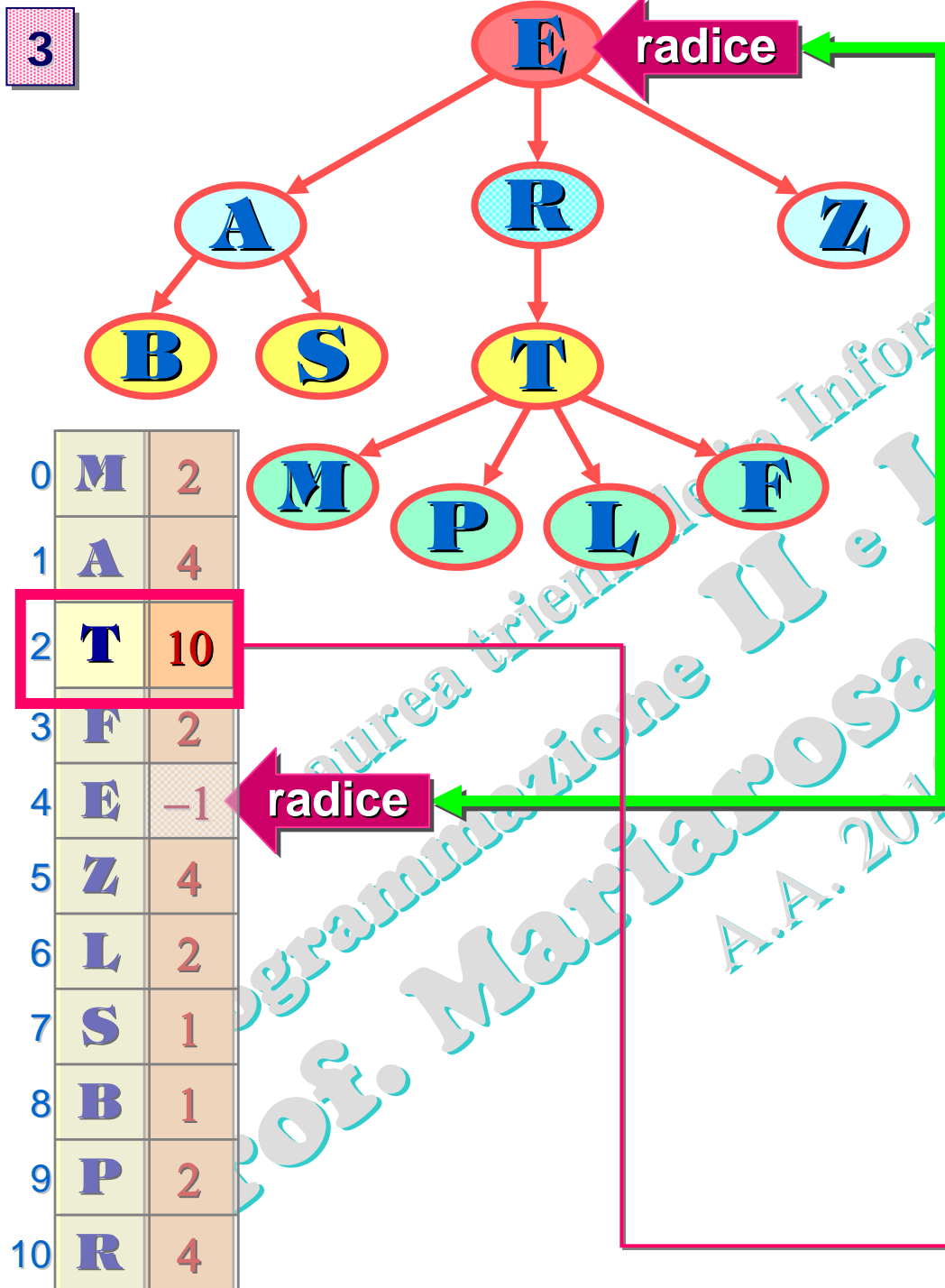
| nodo | | | | |
|--------------|-----------------|-----------------|-----------------|-----------------|
| informazione | | | | |
| grado | pt ₁ | pt ₂ | pt ₃ | pt ₄ |

| | INF | GR | pt ₁ | pt ₂ | pt ₃ | pt ₄ |
|----|-----|----|-----------------|-----------------|-----------------|-----------------|
| 0 | M | 0 | | | | |
| 1 | | 0 | | | | |
| 2 | | 1 | 0 | | | |
| 3 | | 0 | | | | |
| 4 | | 0 | | | | |
| 5 | | 0 | | | | |
| 6 | | 0 | | | | |
| 7 | | 0 | | | | |
| 8 | | 0 | | | | |
| 9 | | 0 | | | | |
| 10 | | 0 | | | | |



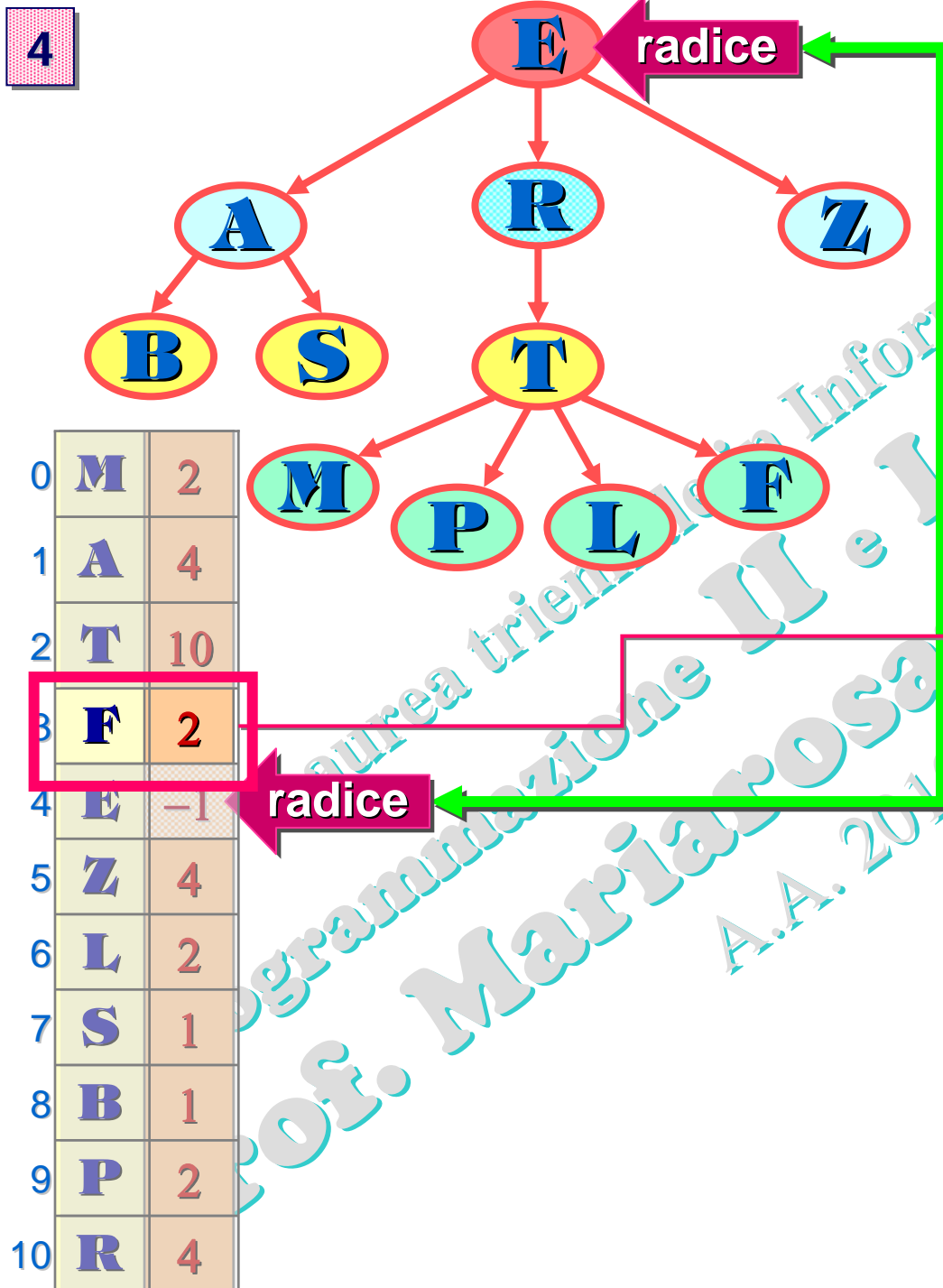
| nodo | | | | |
|--------------|-----------------|-----------------|-----------------|-----------------|
| informazione | | | | |
| grado | pt ₁ | pt ₂ | pt ₃ | pt ₄ |

| | INF | GR | pt ₁ | pt ₂ | pt ₃ | pt ₄ |
|----|-----|----|-----------------|-----------------|-----------------|-----------------|
| 0 | M | 0 | | | | |
| 1 | A | 0 | | | | |
| 2 | | 1 | 0 | | | |
| 3 | | 0 | | | | |
| 4 | | 1 | 1 | | | |
| 5 | | 0 | | | | |
| 6 | | 0 | | | | |
| 7 | | 0 | | | | |
| 8 | | 0 | | | | |
| 9 | | 0 | | | | |
| 10 | | 0 | | | | |



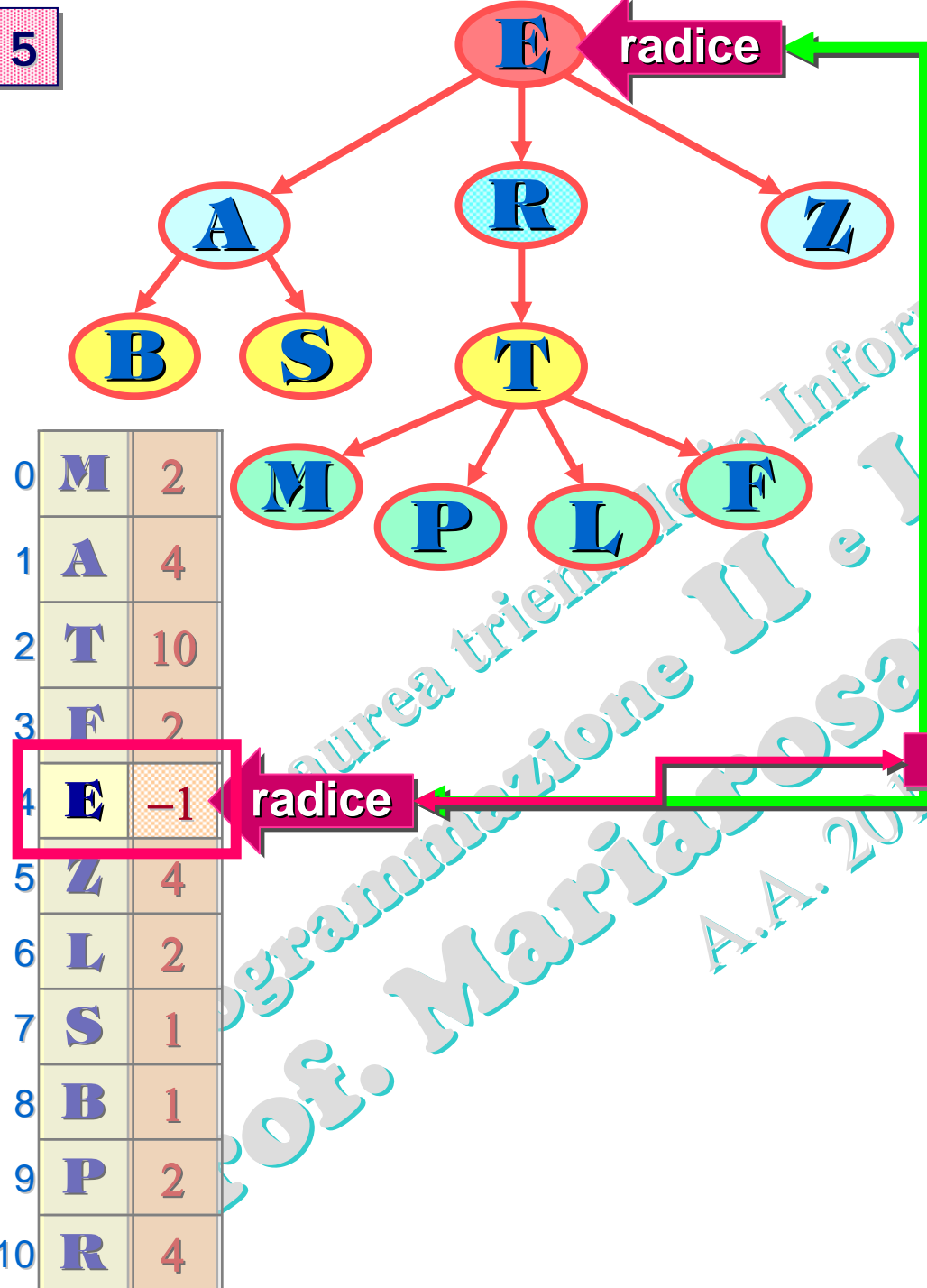
| nodo | | | | |
|--------------|-----------------|-----------------|-----------------|-----------------|
| informazione | | | | |
| grado | pt ₁ | pt ₂ | pt ₃ | pt ₄ |

| INF | GR | pt ₁ | pt ₂ | pt ₃ | pt ₄ |
|-----|----|-----------------|-----------------|-----------------|-----------------|
| 0 | M | 0 | | | |
| 1 | A | 0 | | | |
| 2 | T | 1 | 0 | | |
| 3 | | 0 | | | |
| 4 | | 1 | 1 | | |
| 5 | | 0 | | | |
| 6 | | 0 | | | |
| 7 | | 0 | | | |
| 8 | | 0 | | | |
| 9 | | 0 | | | |
| 10 | | 1 | 2 | | |



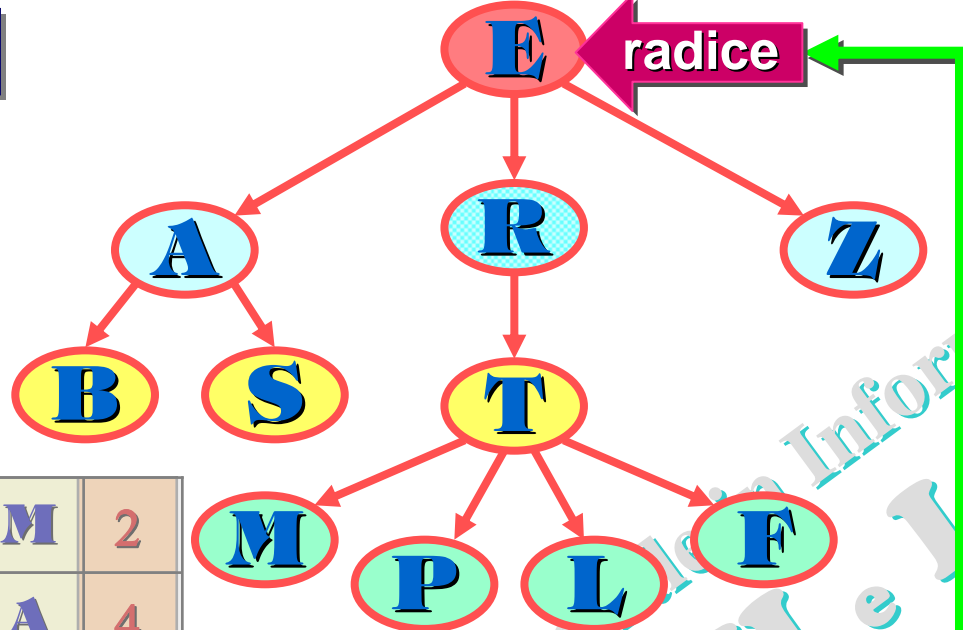
| nodo | | | | |
|--------------|-----------------|-----------------|-----------------|-----------------|
| informazione | | | | |
| grado | pt ₁ | pt ₂ | pt ₃ | pt ₄ |

| | INF | GR | pt ₁ | pt ₂ | pt ₃ | pt ₄ |
|----|-----|----|-----------------|-----------------|-----------------|-----------------|
| 0 | M | 0 | | | | |
| 1 | A | 0 | | | | |
| 2 | T | 2 | 0 | 3 | | |
| 3 | F | 0 | | | | |
| 4 | | 1 | 1 | | | |
| 5 | | 0 | | | | |
| 6 | | 0 | | | | |
| 7 | | 0 | | | | |
| 8 | | 0 | | | | |
| 9 | | 0 | | | | |
| 10 | | 1 | 2 | | | |



| nodo | | | | |
|--------------|-----------------|-----------------|-----------------|-----------------|
| informazione | | | | |
| grado | pt ₁ | pt ₂ | pt ₃ | pt ₄ |

| | INF | GR | pt ₁ | pt ₂ | pt ₃ | pt ₄ |
|----|-----|----|-----------------|-----------------|-----------------|-----------------|
| 0 | M | 0 | | | | |
| 1 | A | 0 | | | | |
| 2 | T | 2 | 0 | 3 | | |
| 3 | F | 0 | | | | |
| 4 | E | 1 | 1 | | | |
| 5 | | 0 | | | | |
| 6 | | 0 | | | | |
| 7 | | 0 | | | | |
| 8 | | 0 | | | | |
| 9 | | 0 | | | | |
| 10 | | 1 | 2 | | | |

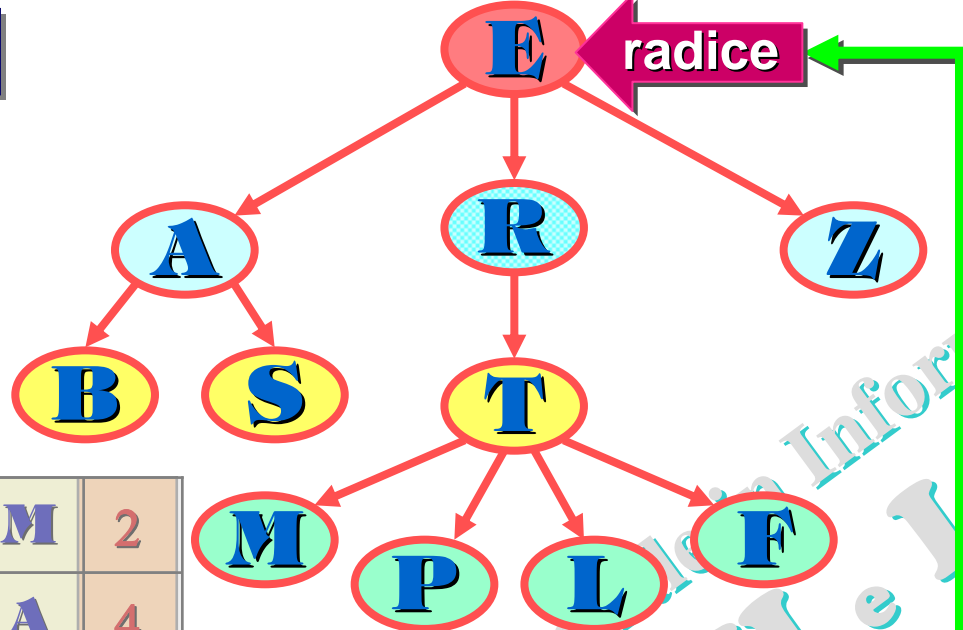


| nodo | | | | |
|--------------|-----------------|-----------------|-----------------|-----------------|
| informazione | | | | |
| grado | pt ₁ | pt ₂ | pt ₃ | pt ₄ |

| | | |
|----|---|----|
| 0 | M | 2 |
| 1 | A | 4 |
| 2 | T | 10 |
| 3 | F | 2 |
| 4 | E | -1 |
| 5 | Z | 4 |
| 6 | L | 2 |
| 7 | S | 1 |
| 8 | B | 1 |
| 9 | P | 2 |
| 10 | R | 4 |

radice

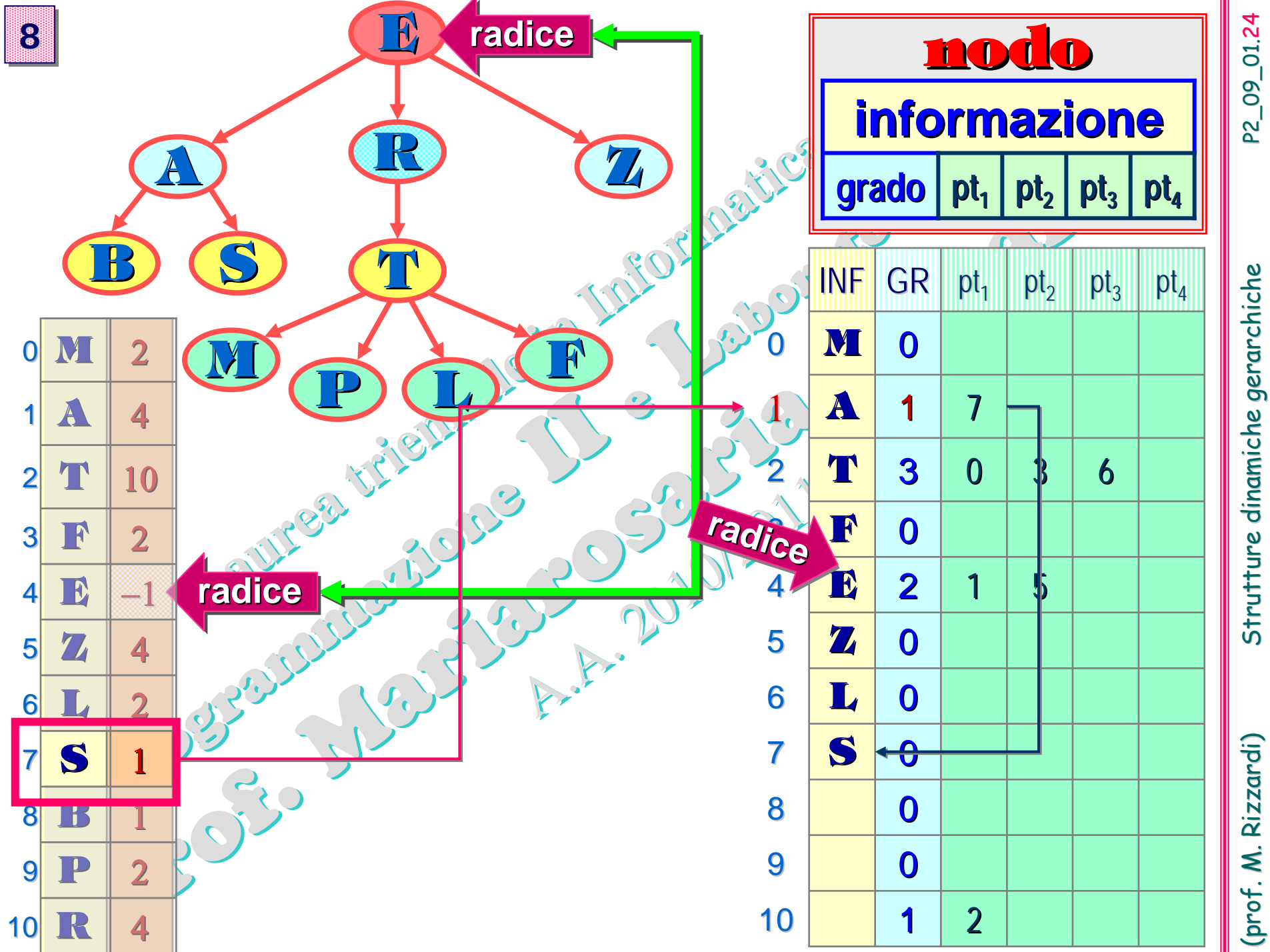
| INF | GR | pt ₁ | pt ₂ | pt ₃ | pt ₄ |
|-----|----|-----------------|-----------------|-----------------|-----------------|
| 0 | M | 0 | | | |
| 1 | A | 0 | | | |
| 2 | T | 2 | 0 | 3 | |
| 3 | F | 0 | | | |
| 4 | E | 2 | 1 | 5 | |
| 5 | Z | 0 | | | |
| 6 | | 0 | | | |
| 7 | | 0 | | | |
| 8 | | 0 | | | |
| 9 | | 0 | | | |
| 10 | | 1 | 2 | | |



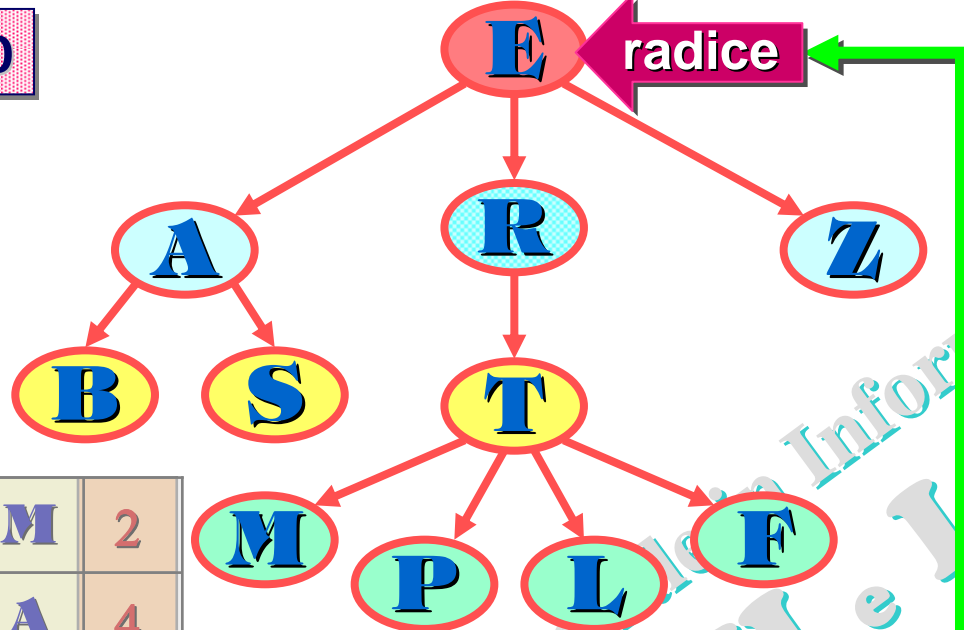
| nodo | | | | |
|--------------|-----------------|-----------------|-----------------|-----------------|
| informazione | | | | |
| grado | pt ₁ | pt ₂ | pt ₃ | pt ₄ |

| | | |
|----|---|----|
| 0 | M | 2 |
| 1 | A | 4 |
| 2 | T | 10 |
| 3 | F | 2 |
| 4 | E | -1 |
| 5 | Z | 4 |
| 6 | L | 2 |
| 7 | S | 1 |
| 8 | B | 1 |
| 9 | P | 2 |
| 10 | R | 4 |

| | INF | GR | pt ₁ | pt ₂ | pt ₃ | pt ₄ |
|----|-----|----|-----------------|-----------------|-----------------|-----------------|
| 0 | M | 0 | | | | |
| 1 | A | 0 | | | | |
| 2 | T | 3 | 0 | 3 | 6 | |
| 3 | F | 0 | | | | |
| 4 | E | 2 | 1 | 5 | | |
| 5 | Z | 0 | | | | |
| 6 | L | 0 | | | | |
| 7 | | 0 | | | | |
| 8 | | 0 | | | | |
| 9 | | 0 | | | | |
| 10 | | 1 | 2 | | | |







| nodo | | | | |
|--------------|-----------------|-----------------|-----------------|-----------------|
| informazione | | | | |
| grado | pt ₁ | pt ₂ | pt ₃ | pt ₄ |

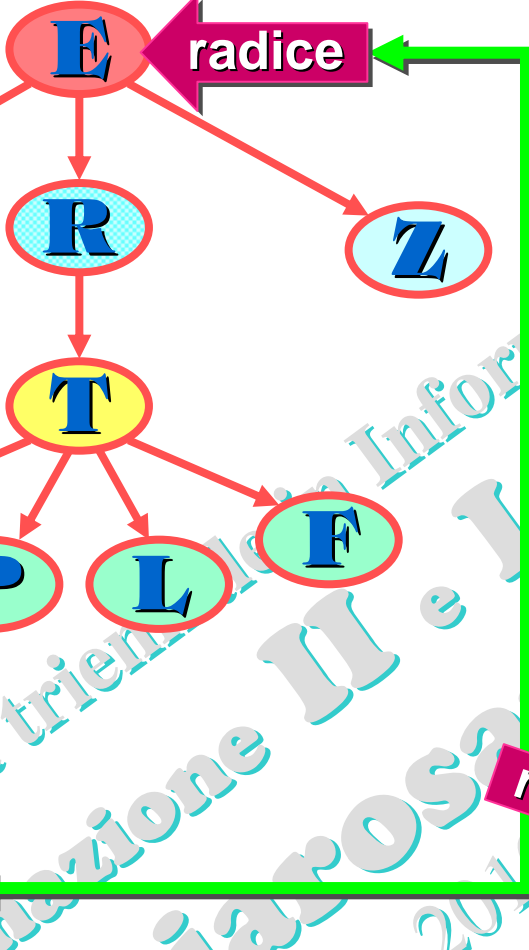
| | | |
|----|---|----|
| 0 | M | 2 |
| 1 | A | 4 |
| 2 | T | 10 |
| 3 | F | 2 |
| 4 | E | -1 |
| 5 | Z | 4 |
| 6 | L | 2 |
| 7 | S | 1 |
| 8 | B | 1 |
| 9 | P | 2 |
| 10 | R | 4 |

| | INF | GR | pt ₁ | pt ₂ | pt ₃ | pt ₄ |
|----|-----|----|-----------------|-----------------|-----------------|-----------------|
| 0 | M | 0 | | | | |
| 1 | A | 2 | 7 | 8 | | |
| 2 | T | 4 | 0 | 3 | 6 | 9 |
| 3 | F | 0 | | | | |
| 4 | E | 2 | 1 | 5 | | |
| 5 | Z | 0 | | | | |
| 6 | L | 0 | | | | |
| 7 | S | 0 | | | | |
| 8 | B | 0 | | | | |
| 9 | P | 0 | | | | |
| 10 | | 1 | 2 | | | |

radice

radice

situazione finale



| nodo | | | | | |
|--------------|-----------------|-----------------|-----------------|-----------------|--|
| informazione | | | | | |
| grado | pt ₁ | pt ₂ | pt ₃ | pt ₄ | |

| | | |
|----|---|----|
| 0 | M | 2 |
| 1 | A | 4 |
| 2 | T | 10 |
| 3 | F | 2 |
| 4 | E | -1 |
| 5 | Z | 4 |
| 6 | L | 2 |
| 7 | S | 1 |
| 8 | B | 1 |
| 9 | P | 2 |
| 10 | R | 4 |

| | INF | GR | pt ₁ | pt ₂ | pt ₃ | pt ₄ |
|----|-----|----|-----------------|-----------------|-----------------|-----------------|
| 0 | M | 0 | | | | |
| 1 | A | 2 | 7 | 8 | | |
| 2 | T | 4 | 0 | 3 | 6 | 9 |
| 3 | F | 0 | | | | |
| 4 | E | 3 | 1 | 5 | 10 | |
| 5 | Z | 0 | | | | |
| 6 | L | 0 | | | | |
| 7 | S | 0 | | | | |
| 8 | B | 0 | | | | |
| 9 | P | 0 | | | | |
| 10 | R | 1 | 2 | | | |

Esercizio:

2

Scrivere *function C* per la costruzione e visita per livelli di un albero (mediante liste multiple). **[liv. 3]**
[suggerimento: i puntatori ai figli di un nodo risiedono in un array dinamicamente allocato e indirizzato dall'unico campo puntatore del nodo dell'albero (`pt_figli`)]

