Esercitazioni di Algoritmi e Strutture Dati

III esercitazione, 17/03/2016

Tong Liu

ESERCIZIO PRECEDENTE

- Es 2.5, Il rango di un elemento di una lista di interi é la somma degli elementi successivi piú se stesso.

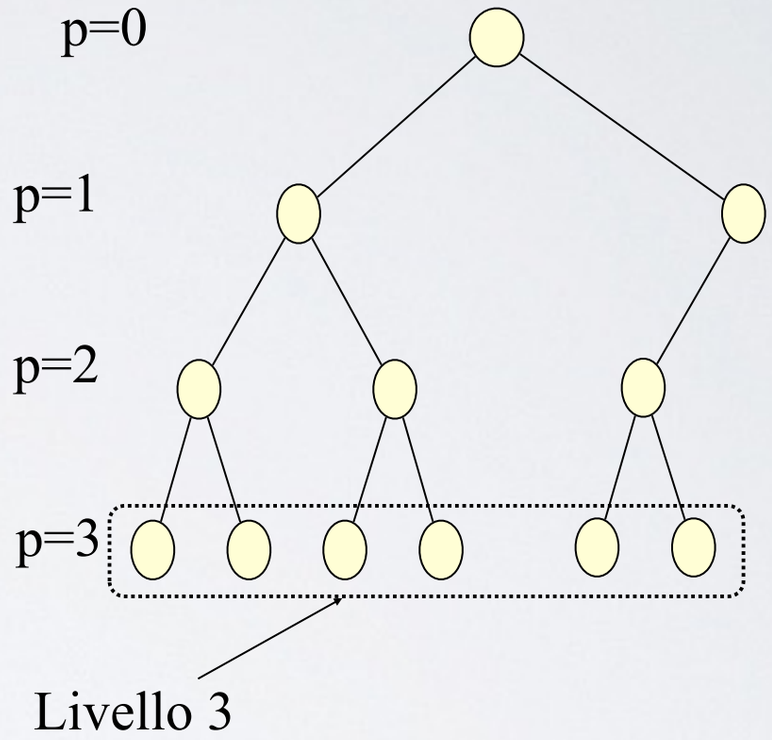
- Rango di [3,2,5] é [10,7,5]

- Creare una funzione che calcola il rango di una lista

ALBERI

ALBERO

- **In un albero**
  - *Profondità* di un nodo: la lunghezza del percorso dalla radice al nodo (i.e., numero archi attraversati)
  - *Livello*: l'insieme dei nodi alla stessa profondità
  - *Altezza* dell'albero: massimo livello delle sue foglie



Altezza albero: 3

ALBERI: OPERAZIONI

TREE

% Costruisce un nuovo albero, costituito da un solo nodo e contenente v
Tree(ITEM v)

% Legge il valore
ITEM read()

% Scrive v nel nodo
write(ITEM v)

% Restituisce il padre; **nil** se questo nodo è radice
TREE parent()

% Restituisce il primo figlio; **nil** se questo nodo è foglia
TREE leftmostChild()

% Restituisce il prossimo fratello del nodo a cui è applicato; **nil** se assente
TREE rightSibling()

% Inserisce il sottoalbero t come primo figlio di questo nodo
insertChild(TREE t)
precondition: t.parent() = **nil**

% Inserisce il sottoalbero t come successivo fratello di questo nodo
insertSibling(TREE t)
precondition: t.parent() = **nil**

% Distrugge il sottoalbero radicato nel primo figlio di questo nodo
deleteChild()

% Distrugge il sottoalbero radicato nel prossimo fratello di questo nodo
deleteSibling()

VISITA : PRE-VISITA

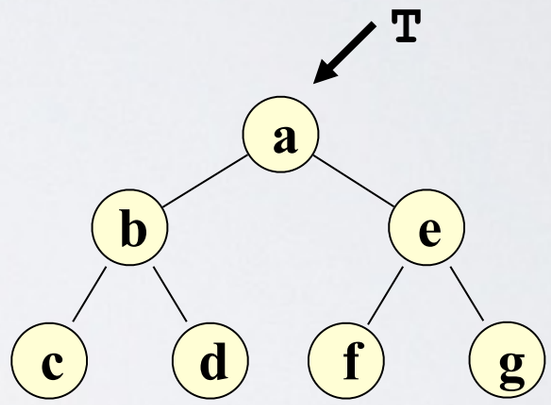
visitaProfondità(TREE t)

precondition: $t \neq$ **nil**

(1) esame "anticipato" del nodo radice di t

```
TREE u ← t.leftmostChild()
while u ≠ nil do
    visitaProfondità(u)
    u ← u.rightSibling()
```

~~(2) esame "posticipato" del nodo radice di t~~



Sequenza: a b c d e f g

VISITA : POST-VISITA

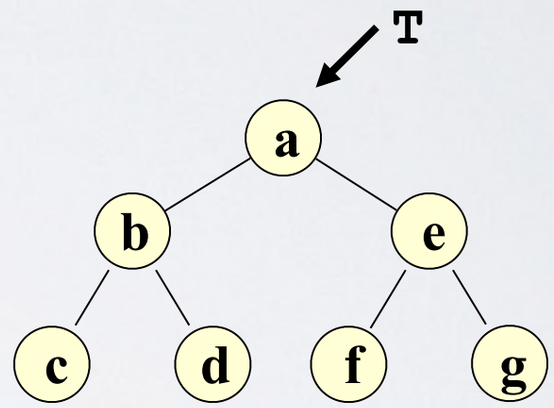
visitaProfondità(TREE t)

precondition: $t \neq$ **nil**

(1) ~~esame "anticipato" del nodo radice di t~~

```
TREE u ← t.leftmostChild()
while u ≠ nil do
    visitaProfondità(u)
    u ← u.rightSibling()
```

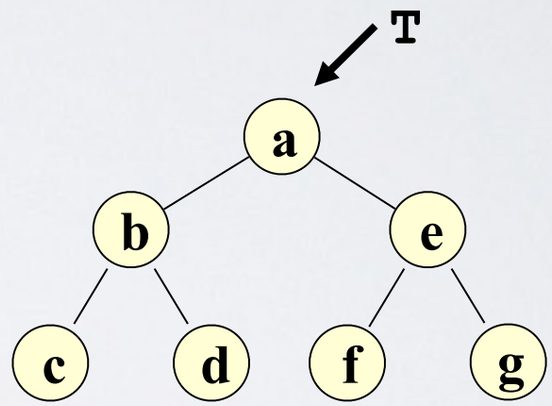
(2) esame "posticipato" del nodo radice di t



Sequenza: `c d b f g e a`

VISITA : IN-VISITA

```
invisita(TREE t)
    precondition: t ≠ nil
    TREE u ← t.leftmostChild()
    integer k ← 0
    while u ≠ nil and k < i do
    |   k ← k + 1
    |   invisita(u)
    |   u ← u.rightSibling()
    esame "simmetrico" del nodo t
    while u ≠ nil do
    |   invisita(u)
    |   u ← u.rightSibling()
```



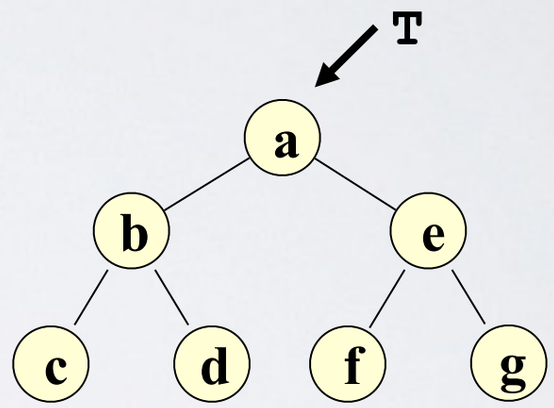
Sequenza (i=1): `c b d a f e g`

VISITA : IN AMPIEZZA

visitaAmpiezza(TREE t)

precondition: $t \neq$ **nil**

QUEUE $Q \leftarrow$ Queue()
Q.enqueue(t)
while not Q.isEmpty() **do**
  TREE $u \leftarrow Q$.dequeue()
  esame "per livelli" del nodo u
  $u \leftarrow u$.leftmostChild()
  while $u \neq$ **nil do**
    Q.enqueue(u)
    $u \leftarrow u$.rightSibling()



Sequenza: a b e c d f g

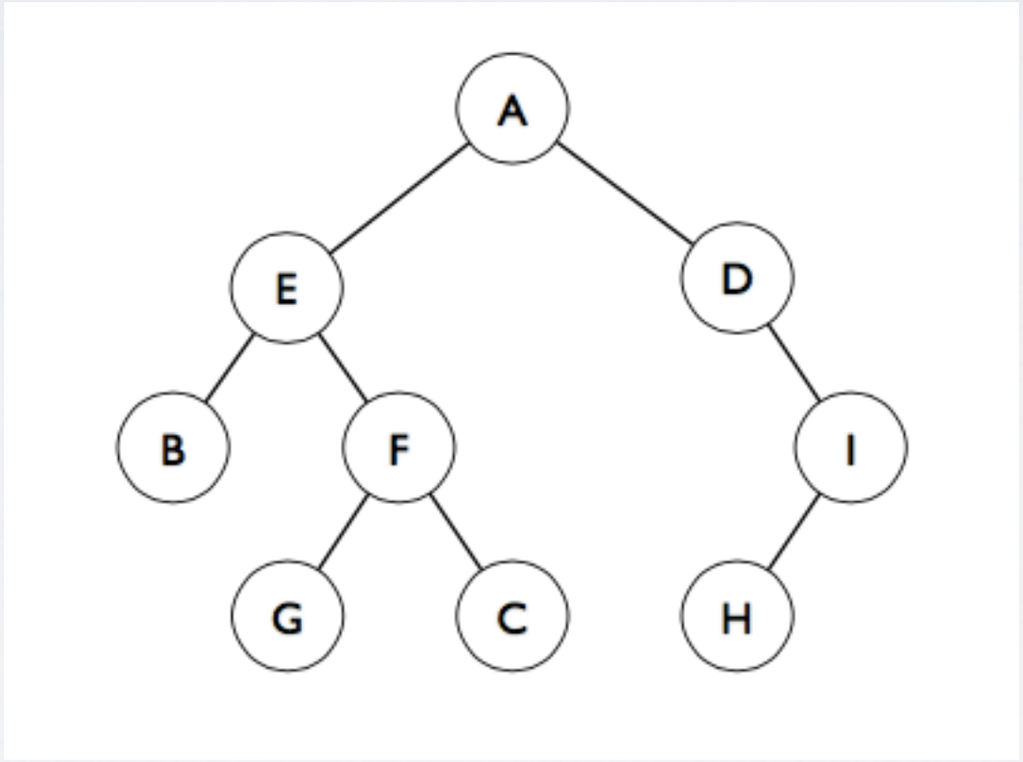
ESERCIZIO VISITE

[Libro 5.4] Gli ordini di visita di un albero binario di 9 nodi sono i seguenti:

- A, E, B, F, G, C, D, I, H(anticipato)
- B, G, C, F, E, H, I, D, A (posticipato)
- B, E, G, F, C, A, D, H, I (simmetrico).

Si ricostruisca l'albero binario e si illustri *brevemente* il ragionamento.

SOLUZIONE



ESERCIZIO ALTEZZA

[Libro 5.1] L'altezza di un albero ordinato è il massimo livello delle sue foglie. Si fornisca una funzione che calcoli in tempo ottimo l'altezza di un albero ordinato T di n nodi.

ESERCIZIO CANCELLA

[Libro 5.2] Dato un albero ordinato i cui nodi contengono valori interi, se ne vogliono cancellare tutte le foglie per le quali il percorso radice-foglia ha somma complessiva dei valori uguale a *k*. Fornire una procedura di complessità ottima.

(Suggerimento: utilizzare l'algoritmo pre-visita, modificare tale funzione in modo che la funzione ritorna un bool per dare segnale di cancellazione)

SOLUZIONI

Le soluzioni si trovano nella pagina 104 del libro: Alan Bertossi, Alberto Montresor. Algoritmi e Strutture di Dati, 3a edizione. Città Studi Edizioni, 2014

LINKS

Visita Albero: https://www.youtube.com/watch?v=tYPUvEZF8XE

Approfondimenti: https://en.wikipedia.org/wiki/Tree_traversal