LABORATORIO DI PROGRAMMAZIONE 2
Corso di laurea in matematica

Cenni agli alberi

Marco Lapegna
Dipartimento di Matematica e Applicazioni
Universita' degli Studi di Napoli Federico II

wpage.unina.it/lapegna

gli alberi

Un **albero** è una struttura dati **NON LINEARE** organizzata gerarchicamente.

È costituito da un insieme di **nodi** collegati tra di loro:

- ogni nodo contiene almeno una chiave (key, label, …) e 1 o piu' puntatori a nodi figli
- i nodi collegati ad un certo nodo vengono detti **figli** del nodo
- i nodi che non hanno figli sono detti **foglie** dell'albero
- c' è un unico nodo che non è figlio di alcun altro nodo, detto **radice (root)** dell'albero

(scendendo a partire dalla radice si raggiungono tutti i nodi dell'albero)

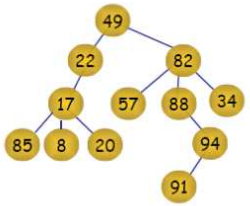
un esempio

Albero in cui le *label* (etichette) sono degli interi

La radice ha label 49

I figli della radice hanno label 22 e 82

Le foglie hanno label 85, 8, 20, 57, 91, 34



un esempio di albero

alcune definizioni

Cammino in un albero:
sequenza di nodi, in cui ogni nodo è figlio del nodo che lo precede nella sequenza

Livello (o **profondità**) di un nodo è la sua distanza dalla radice (quanto "in basso" si trova nell'albero).

- Definizione formale (induttiva) di livello di un nodo:
  - la radice ha livello ***0***
  - se un nodo ha livello ***i***, allora i suoi figli hanno livello ***i + 1***

Il **livello i** di un albero è formato da tutti i nodi a livello ***i***.

alberi binari

una struttura dati molto comune e'
l'albero binario

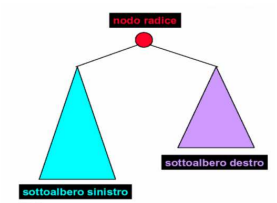
Un **albero binario** è un albero in cui ogni nodo ha **al massimo 2 figli,** chiamati

- **figlio sinistro**
- **figlio destro**

se consideriamo la parte di albero costituita dal figlio sinistro e tutti i suoi ''discendenti'', questa è di nuovo unalbero binario, detto **sottoalbero sinistro**

(analogamente si definisce il **sottoalbero destro**)

L'albero è una **struttura ricorsiva** non lineare.



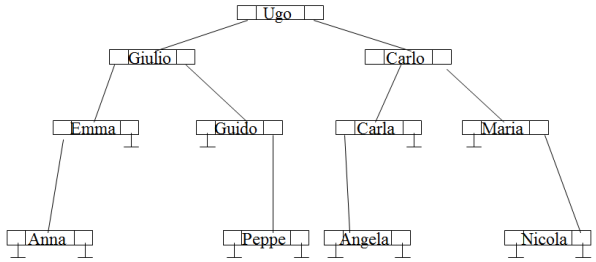
struttura di un albero binario

visita di un albero binario

3 modalita' di visita:

- **Symmetric order**. Si visitano ricorsivamente nell'ordine
  - il sottoalbero sinistro
  - la radice
  - il sottoalbero destro
- **Pre order**. Si visitano ricorsivamente nell'ordine
  - la radice
  - il sottoalbero sinistro
  - il sottoalbero destro
- **Post order**. Si visitano ricorsivamente nell'ordine
  - il sottoalbero sinistro
  - il sottoalbero destro
  - la radice

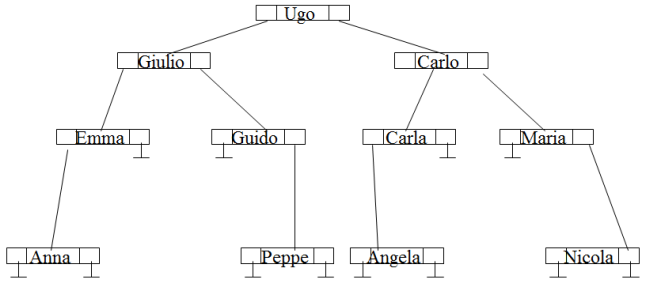
symmetric order



symmetric order:

Anna, Emma, Giulio, Guido, Peppe, Ugo, Angela, Carla, Carlo, Maria, Nicola

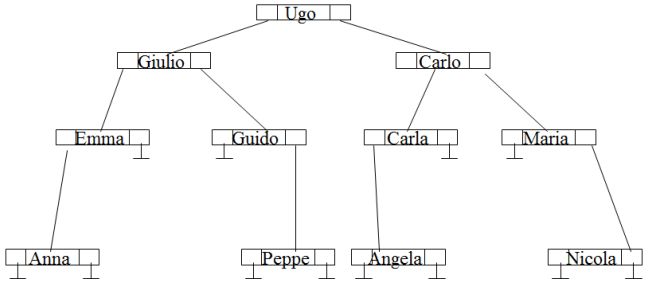
pre order



***pre order*:**

Ugo, Giulio, Emma, Anna, Guido, Peppe, Carlo, Carla, Angela, Maria, Nicola

post order



post order

Anna, Emma, Peppe, Guido, Giulio, Angela, Carla, Nicola, Maria, Carlo, Ugo

procedura symmetric order

per la visita con la modalita' symmetric order
e' possibile usare una semplice procedura ricorsiva

```
procedure symmetric_order (in: root)
      if (root!= NULL) then
          symmetric_order(root.left)
          print root.info
          symmetric_order(root.right)
      endif
end procedure
```


procedura pre order

per la visita con la modalita' pre order
e' possibile usare una semplice procedura ricorsiva

```
procedure pre_order (in: root)
      if (root!= NULL) then
          print root.info
          pre_order(root.left)
          pre_order(root.right)
      endif
end procedure
```


procedura post order

per la visita con la modalita' pre order
e' possibile usare una semplice procedura ricorsiva

```
procedure post_order (in: root)
      if (root!= NULL) then
          post_order(root.left)
          post_order(root.right)
          print root.info
      endif
end procedure
```

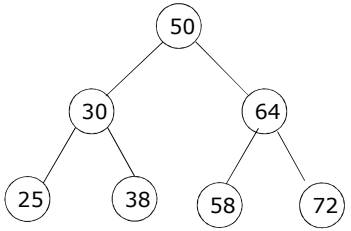
ordinamento negli alberi

L'albero e' una struttura non lineare.

E' possibile definire un ordinamento ?

ALBERI BINARI DI RICERCA

- Per ogni nodo **n** dell'albero: tutte le chiavi dei nodi contenuti nel sottoalbero sinistro di **n** hanno valore **strettamente minore** della chiave contenuta in **n**,
- tutte le chiavi dei nodi contenuti nel sottoalbero destro di **n** hanno valore **strettamente maggiore** della chiave contenuta in **n**.

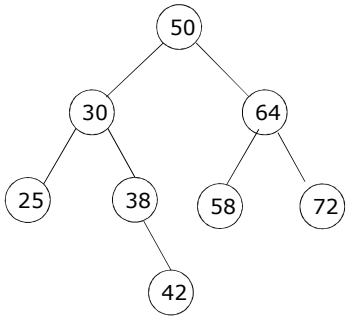


Osservazione: La visita di un albero binario di ricerca con il symmetric order fornisce gli elementi in ordine crescente

inserimento ricorsivo

caso banale: albero vuoto
il nodo da inserire e' diventa la radice dell'albero

caso generale: si vede se inserire il nodo a sinistra oppure a destra della radice



inserimento di 42

costruire un albero binario di ricerca

```
procedura inserisci(in: radice, e)

if(radice==NULL) then                    // albero vuoto
        crea nodo
        nodo.info=e;
        nodo.left=NULL;
        nodo.right=NULL;
        radice=&aux;
else

        if( radice. info > e) then
                inserisci(radice.left, e);
        else
                inserisci(radice.right, e);
        endif
endif

end procedure
```

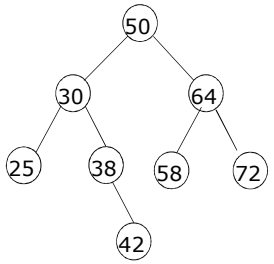

ricerca in un albero binario di ricerca

```
procedura ricerca(in: radice, e; out T)

if(radice==NULL) then                    // albero vuoto
        T = falso
else

        if( radice. info == e) then
                T = vero
                else
                        if (radice.info > e) then
                                ricerca(in: radice.left, e; out: T);
                        else
                                ricerca(in: radice.right, e; out: T);
                        endif
                endif
        endif
endif
end procedure
```

complessita' di tempo

La ricerca di un elemento in un albero con N nodi parte dalla radice e va verso le foglie

Se l'albero non e' sbilanciato la complessita' di tempo e' quindi uguale al
numero di livelli +1
presenti nell'albero



livello 0

livello 1

livello 2

livello 4

$T(N) = \log_2(N)$