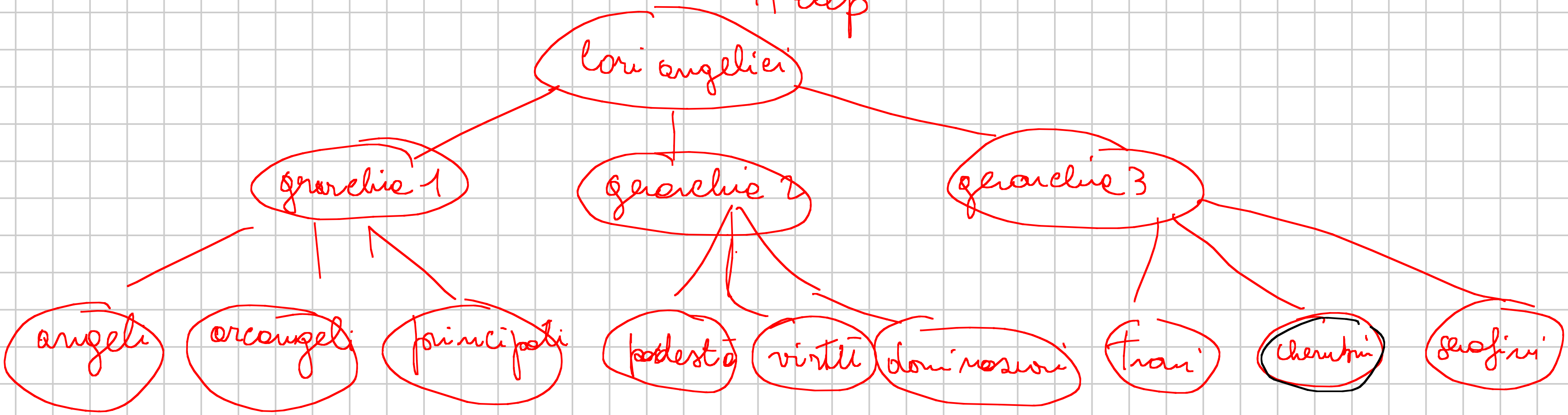
- Alberi di decisione per il calcolo dei limiti inf.
- Alberi di ricorsione per lo studio di eq. di ricorrenze e per la simulaz. di algoritmi Divide et Impera
- Albero come struttura dati (per rendere più efficienti operazioni)

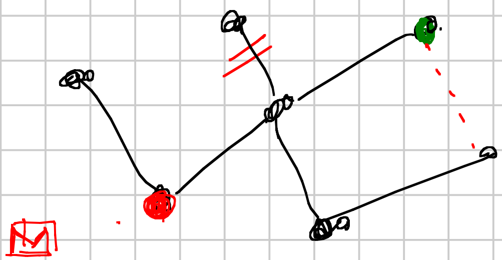
Heap



Alberi per rappresentare informazione di per sé gerarchica

Alberi come reti di connessione

albero libero



Albero è la connessione globale col numero minimo di connessioni tra i nodi

$|E| = |V| - 1$ proprietà vera per ogni albero

Shortest Paths Tree

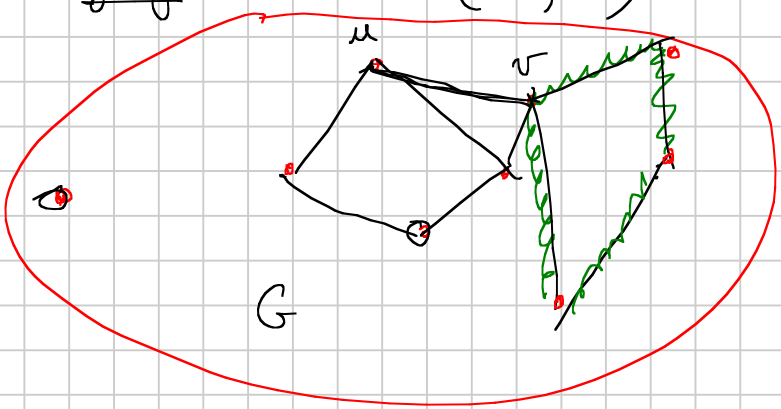
Routing di messaggi

Alberi in natura

Albero: è un grafo connesso privo di cicli

Grafo $G = (V, E)$ Vertices (nodes) Edges (Arcs)

Vertici, nodi Spigoli, archi



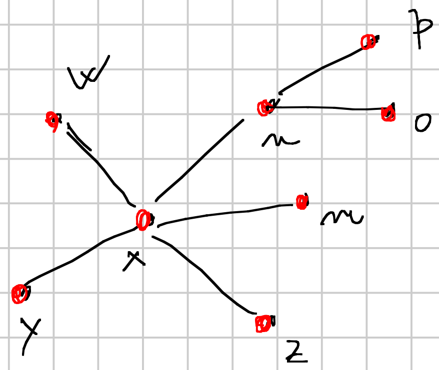
$|V| = n$

$|E| = m$ di archi (u, v)

G è connesso se tutti i suoi vertici sono raggiungibili

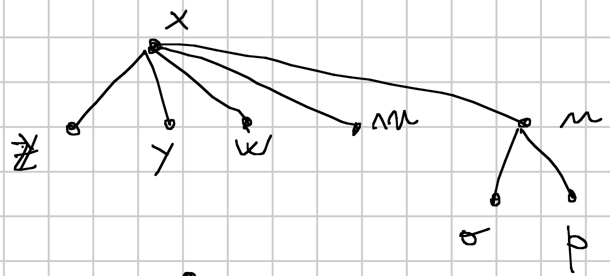
un ciclo un percorso formato da archi che inizia e termina in un vertice.

albero libero



$|E| = |V| - 1$

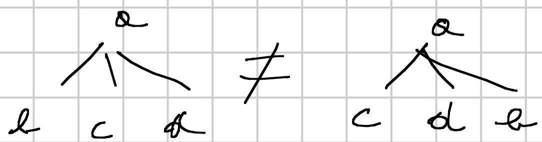
selezione di x come radice



i nodi nel livello possono avere un ordinamento o meno

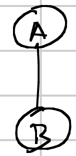
$\Downarrow$

alberi ordinati



Titolo nota 15/04/2016

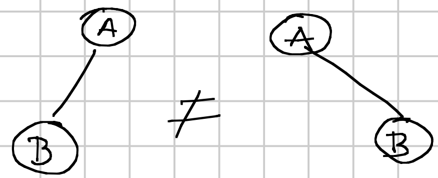
alberi binari (alberi n-ari)



<u>Def. ricorsiva</u>

- vuoto
- i suoi nodi si possono dividere in 3 sottoinsiemi
  - radice
  - sottoalbero sinistro
  - sottoalbero destro

} alberi binari

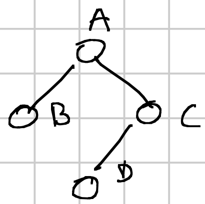


B è figlio sinistro

B è figlio destro

Rappresentazione in memoria di alberi binari n nodi.

1) Rappresentazione in array di n elementi



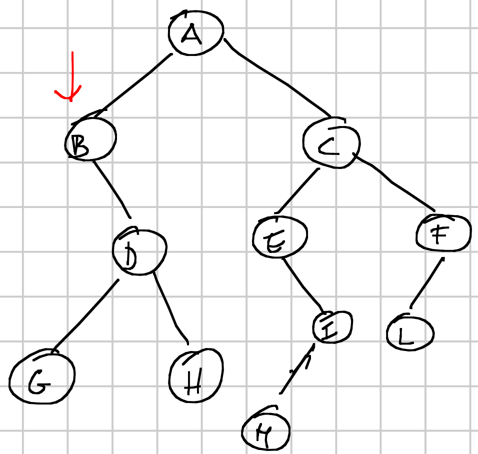
A	B	C	$\emptyset$	$\emptyset$	D
1	2	3	4	5	6

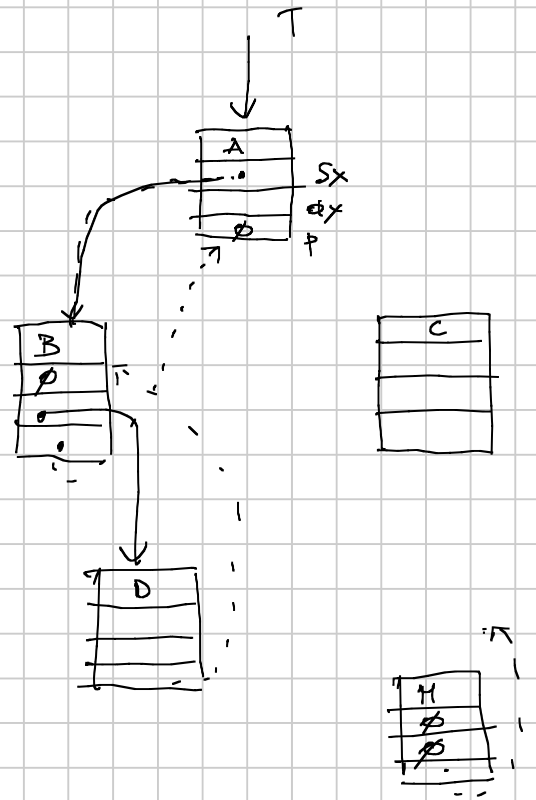
uso inefficiente della memoria.

caso pessimo: spreco esponenziale in n

non si usa a meno di avere informazioni molto precise sull'albero (come Heap)

2) Con puntatori





nodo	Key
punt. sin	sx
punt. des	dx
punt padre	px

T. Key = A

T = T.sx

Operazioni tipiche su alberi binari

1) Dimensione dell'albero (numero dei nodi)

L'albero è conosciuto all'esterno solo col puntatore alla radice.

2) Profondità di un nodo v = distanza del nodo dalla radice

3) Altezza di un albero (o di un nodo) = distanza della radice in numero di archi dalla foglia più lontana

Divide et Impera sugli alberi binari

```
Dimensione (u):
if (u == NULL) return 0;
else {
    Dimsx = Dimensione (u.sx);
    Dimdx = Dimensione (u.dx);
    dim = Dimsx + Dimdx + 1;
    return dim;
}
```

Terminazione

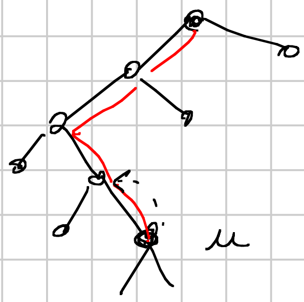
ricorsione

combinazione

Determinare la profondità di un nodo di puntatore u

```
Profondità (u);
p = 0;
while (u.px != NULL) {
    p = p + 1;
    u = u.px
}
```

finché non raggiungo la radice



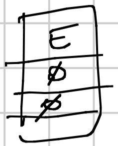
Calcola l'altezza della radice dell'albero di puntatore u con metodo divide et impera

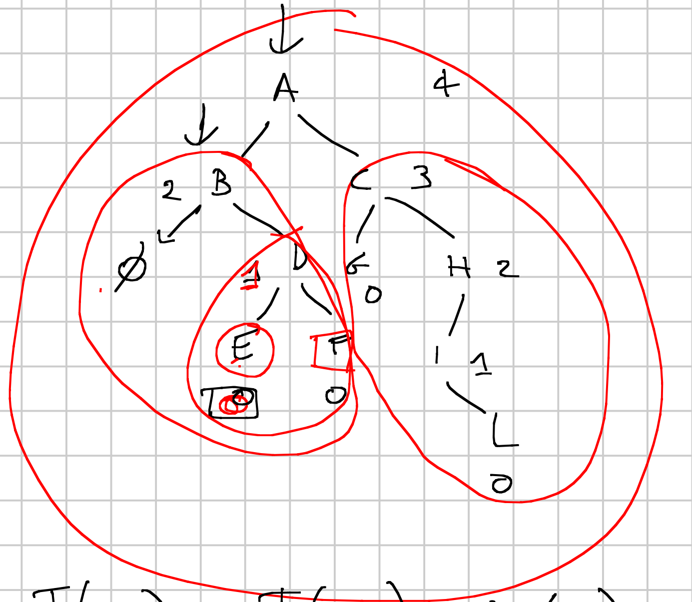
```
Altezza (u)

if (u == NULL)
        return -1;
else {
-  alt_sx = Altezza (u.sx);
-  alt_dx = Altezza (u.dx);
-  alt = max (alt_sx, alt_dx) + 1;
   return alt;
}
```

terminazione

ricorsione

ricombinazione





$T(n) = 2T\left(\frac{n}{2}\right) + \cancel{\Theta(1)} = \Theta(n)$

$T(n) = T(n_s) + T(n_d) + \Theta(1)$

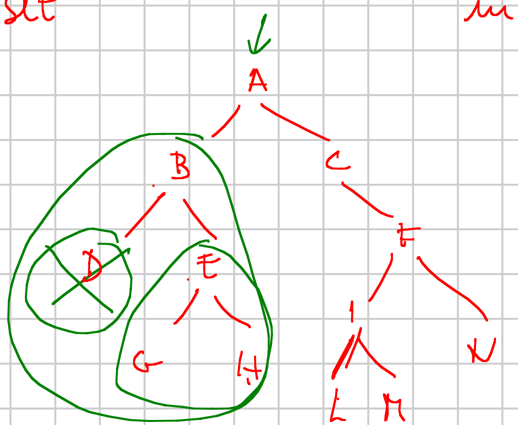
Recuperare tutta l'informazione associata all'albero

Visite Visit

in ordine anticipato
pre-visit

in ordine simmetrico
in-visit

in ordine posticipato o differito
post-visit



Visita in ordine anticipato

if l'albero non è vuoto

1) Esamina la radice

2) Visita il sott. sin in ordine anticipato

3) Visita il sott. destro "

A B D E G H C F I L M N

D B G E H A C L I M F N

D G H E B L M I N F C A

```
IN-Visit (u)
if (u != NULL) {
    IN-Visit (u.sx);
    Print (u.Key);
    IN-Visit (u.dx);
}
```

Visita in ordine simmetrico

```
POST-VISIT(u);
if (u != NULL) {
    POST-Visit (u.sx);
    POST-Visit (u.dx);
    Print (u.Key);
}
```

Visita in ordine differito

$$T(n) = T(n_s) + T(n_d) + 1$$

n_s e n_d = numero di nodi del sottoalbero sinistro e del sott. destro

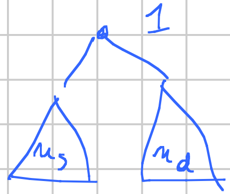
$$\boxed{T(n) = n}$$

prova per induzione

base ; $n = 1$ $\quad T(1) = 1$ vero

ipotesi induttiva ; posto vero per tutti gli $n' < n$

$$T(n) = n_s + n_d + 1 = n$$



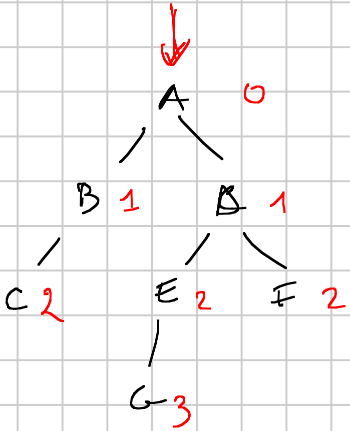
$n = n_s + n_d + 1$

Algoritmi su alberi "tipo" Divide et Impera richiedono tempo lineare

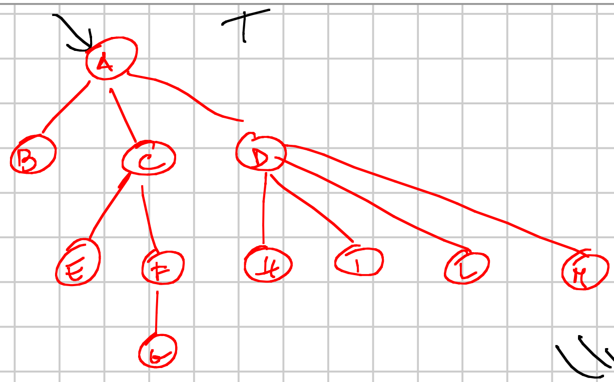
Esercizio ; calcolare il numero di foglie di un albero binario di punt. r.

<u>Esercizio</u>

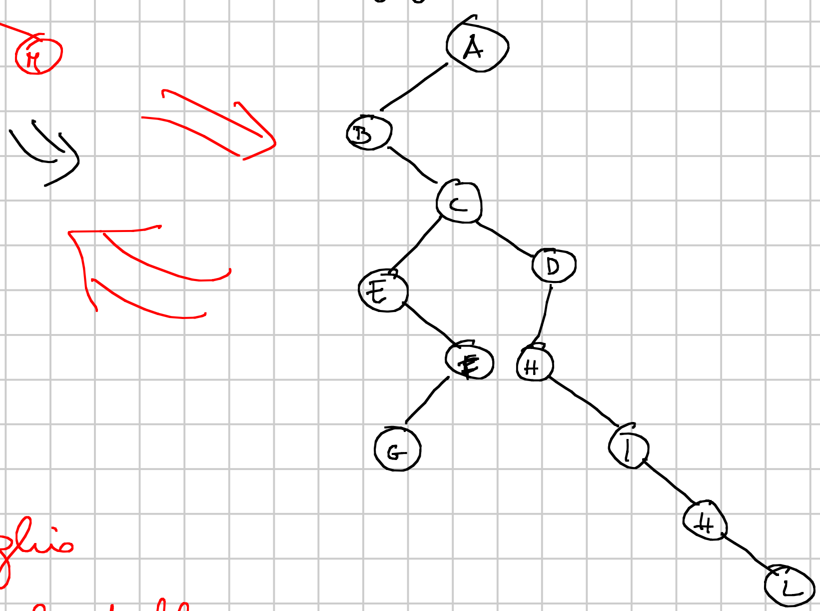
Calcola la profondità di tutti i nodi dell'albero di puntatore u.



Rappresentazione alberi



non sappiamo il n° max di figli



Regola di trasformazione di albero in albero binario

Le radici corrispondono

∀ nodo

il figlio sinistro è il primo figlio

il figlio destro è il successivo fratello