

ESERCIZIO 1 (Alberi binari)

Progettare un algoritmo che verifichi se un albero binario è COMPLETO

↓
ogni nodo interno ha esattamente 2 figli.

Completo(u)

```
if (u == NIL || (u.left == NIL && u.right == NIL))
```

```
    return TRUE;
```

```
if ((u.left == NIL && u.right != NIL) ||  
    (u.left != NIL && u.right == NIL))
```

```
    return FALSE;
```

```
if (!Completo(u.left)) return FALSE;
```

```
return Completo(u.right);
```

$$T(n) = O(n)$$

Albero ordinabile
rappresentato in forma binaria.

foglie

Foglie(u)

if (u == NIL) return 0;

if (u.left-child == NIL)

return 1 + Foglie(u.right-sibling);

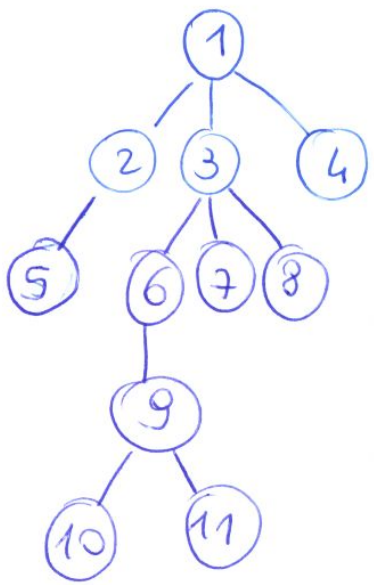
else return Foglie(u.left-child) +
Foglie(u.right-sibling)

$$T(n) = \Theta(n)$$

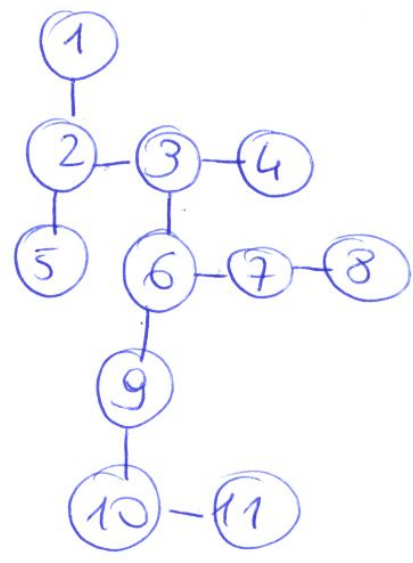
le foglie sono i nodi il cui puntatore left-child
è NIL

Albero ordinale
 rappresentato in forma binaria

Altezza



albero ordinale,
 $h = 4$



forma binaria
 (albero di altezza 6)

Altezza (u)

```

if (u == NIL) return -1;
altFiglio = Altezza (u.left-child);
altFratello = Altezza (u.right-sibling);
altezza = max { 1 + altFiglio, altFratello }
return altezza;
  
```

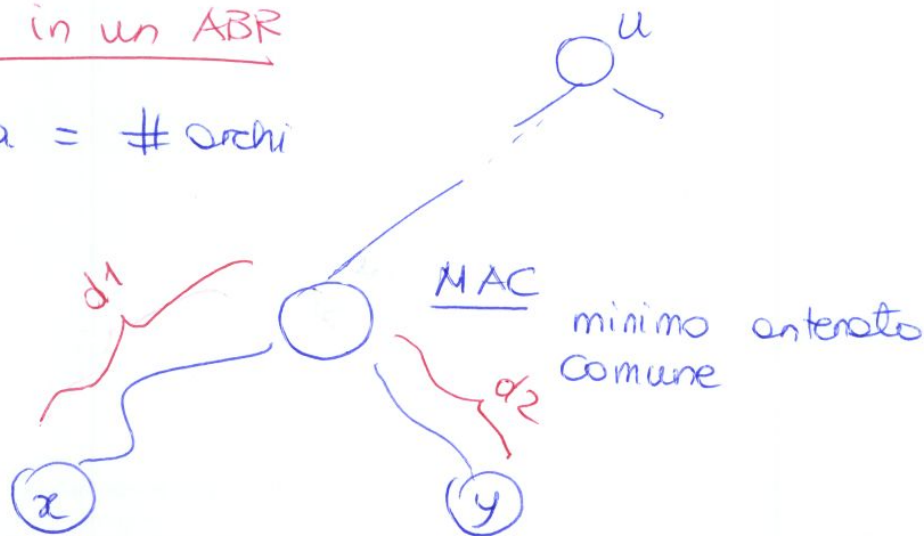
$T(n) = \Theta(n)$

* il fratello è sullo stesso livello. L'altezza non deve essere incrementata *

ESERCIZIO 6

Distanza in un ABR

distanza = # archi



occorre trovare il nodo più vicino ai nodi di chiave x e y che è antenato di entrambi.

TROVA-MAC (u, x, y) // $x < y$

if ($u.key \geq x$) {

if ($u.key \leq y$) return u ;

else return TROVA-MAC ($u.left, x, y$);

}

else return TROVA-MAC ($u.right, x, y$);

Una volta trovata il ~~MAC~~ MAC, si contano (e infine si sommano) le distanze tra i nodi di chiave x e y e il MAC

$$\text{distanza}(x, y) = \text{distanza}(\text{MAC}, x) + \text{distanza}(\text{MAC}, y)$$

~~la chiave del MAC~~

Distanza (u, x, y) // u : radice Albero
// x, y : chiavi $\in T_u$

MAC = TROVA-MAC (u, x, y);

d_1 = Controlli (MAC, x);

d_2 = Controlli (MAC, y);

return $d_1 + d_2$;

Controlli (u, k) // u : nodo ($\neq NIL$)
// k : chiave $\in T_u$

if ($u.key == k$) return 0;

if ($u.key > k$) return $1 +$ Controlli ($u.left, k$)

else return $1 +$ Controlli ($u.right, k$)

$$T(n) = O(h)$$

Mediano

Dato un ABR, stabilire se il nodo u è il nodo mediano del sottalbero di cui è radice.

Mediano (u)

if ($u == \text{null}$) return true;

$s = \text{Dim}(u.\text{left});$

$d = \text{Dim}(u.\text{right});$

return ($|s - d| \leq 1$);

$$T(n) = \Theta(n)$$

Dim (u)

if ($u == \text{null}$) return 0;

else return $1 + \text{Dim}(u.\text{left}) + \text{Dim}(u.\text{right});$