

Algoritmi e Strutture Dati

Lezione 27

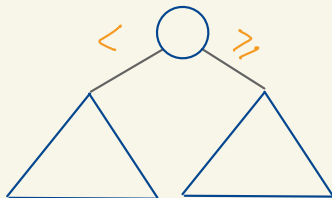
30 novembre 2022

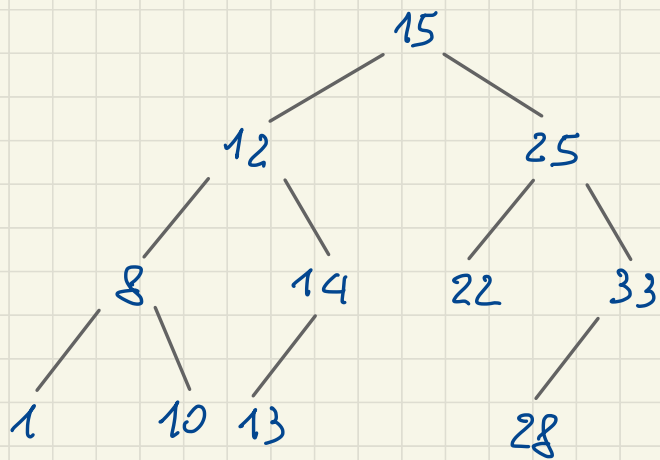
Alberi di ricerca

Alberi binari di ricerca

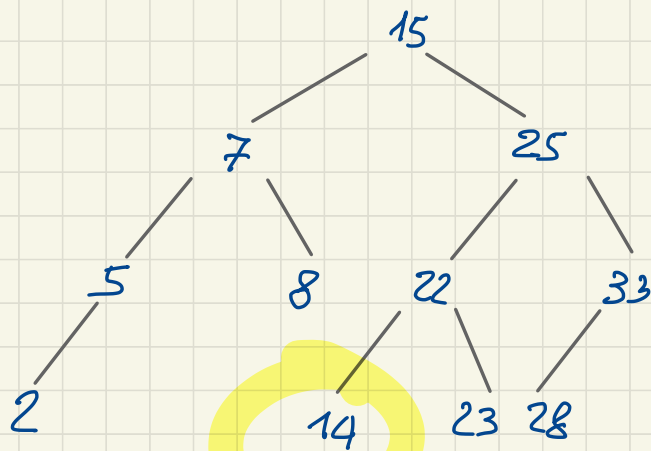
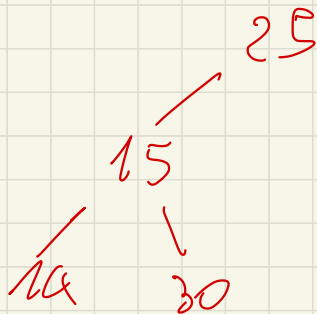
Un *albero binario di ricerca* è un albero binario in cui *per ogni nodo n* :

- il valore di ogni chiave contenuta nel **sottoalbero sinistro** di n è **minore** della chiave contenuta in n ,
- il valore di ogni chiave contenuta nel **sottoalbero destro** di n è **maggiore o uguale** della chiave contenuta in n .





SI

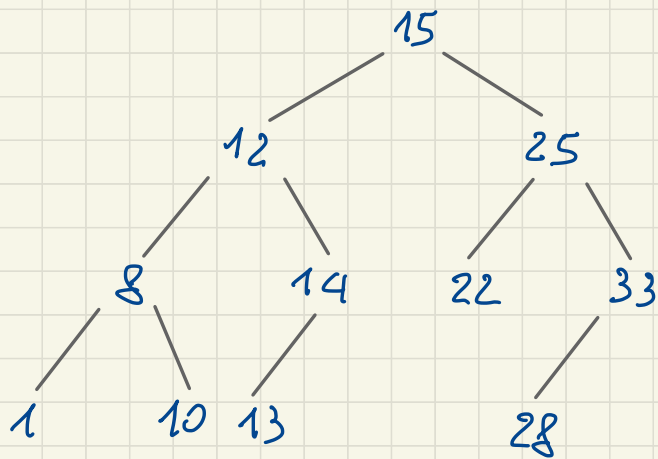


No!

dato {

chiave	
altri campi	
Sx	dx

accesso ai
sottoalberi

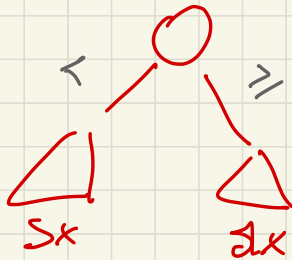


Anfragen

15, 12, 25, 8, 14, 22, 33, 1, 10
13, 28

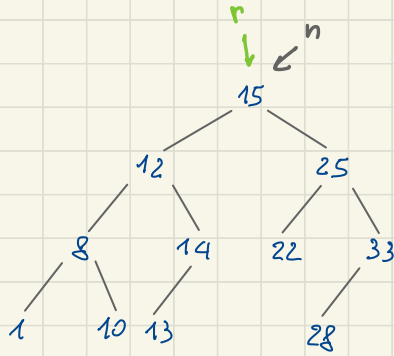
Profundität ordnungssymmetrie

1 8 10 12 13 14 15 22 25 28 33



S_x ; P , d_x

Trovare il nodo con chiave max (o min)



FUNZIONE massimo (Albero Ricorre r) \rightarrow Nodo

IF $r = \text{null}$ THEN

RETURN null

ELSE

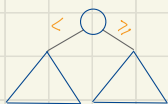
$n \leftarrow r$

WHILE $n.\text{dx} \neq \text{null}$ DO

$n \leftarrow n.\text{dx}$

RETURN n

Ricerca ricorsiva



FUNZIONE trova (Albero Ricerca r , tipo Chiave k) \rightarrow Nodo

IF $r = \text{null}$ THEN

RETURN null

ELSE IF $k < r.\text{chiave}$ THEN

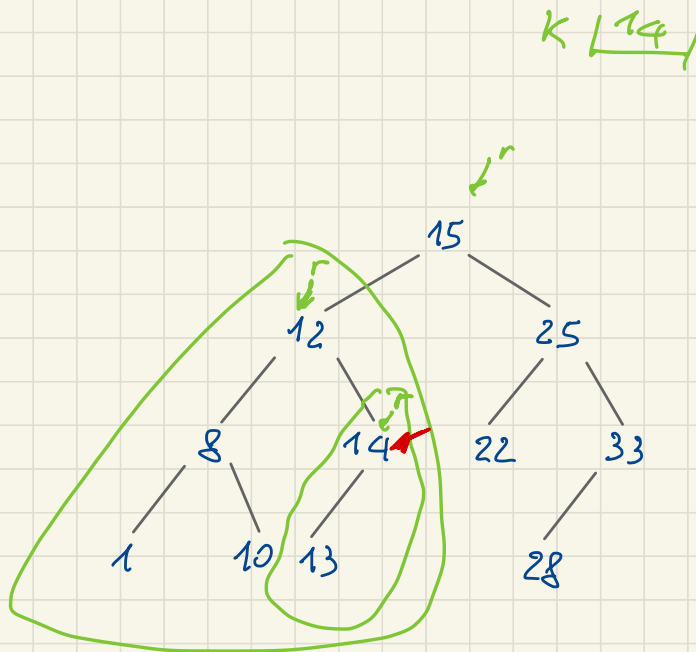
RETURN trova($r.\text{sx}$, k)

ELSE IF $k > r.\text{chiave}$ THEN

RETURN trova($r.\text{dx}$, k)

ELSE RETURN r

ricorsioni in coda



Ricerca iterativa

FUNZIONE trova (AlberoRicerca n, tipoChiave k)
→ Nodo

$n \leftarrow r$

WHILE $n \neq null$ AND $n.chiave \neq k$ DO

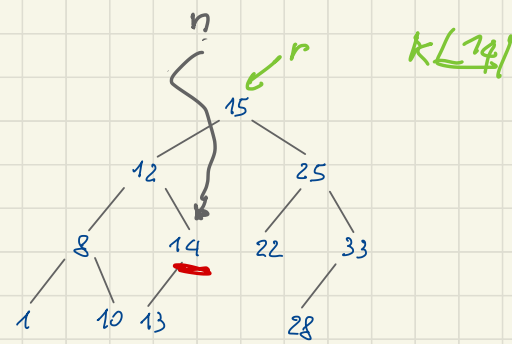
IF $k < n.chiave$ THEN

$n \leftarrow n.sx$

ELSE

$n \leftarrow n.sr$

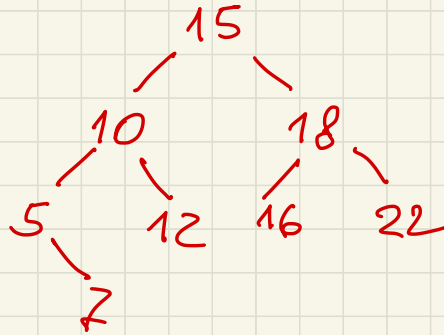
RETURN n



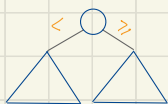
Inserimento

Disegnare l'albero di ricerca che si ottiene a partire da un albero vuoto inserendo uno dopo l'altro, nell'ordine indicato, i seguenti numeri:

15 10 5 18 12 7 16 22



Inserimento ricorsivo



albero vuoto \rightarrow inserisci alla radice

FUNZIONE inserisci (Albero Ricerca r , elemento d) \rightarrow Albero Ricerca

$k \leftarrow d.$ chiave

IF $r = \text{null}$ THEN

$r \leftarrow$ riferimento a nuovo nodo

$r.$ chiave $\leftarrow k$

$r.$ altri campi $\leftarrow d.$ altri campi

$r.sx \leftarrow \text{null}$

$r.dx \leftarrow \text{null}$

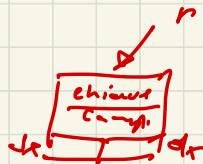
ELSE IF $k < r.$ chiave THEN

$r.sx \leftarrow \text{inserisci}(r.sx, d)$

ELSE

$r.dx \leftarrow \text{inserisci}(r.dx, d)$

RETURN r



FUNZIONE inserisci (AlberoRicerca r, elemento d) \rightarrow AlberoRicerca

k \leftarrow d.chiave

IF r = null THEN

r \leftarrow riferimento a nuovo nodo

r.chiave \leftarrow k

r.altriCampi \leftarrow d.altriCampi

r.sx \leftarrow null

r.dx \leftarrow null

ELSE IF k < r.chiave THEN

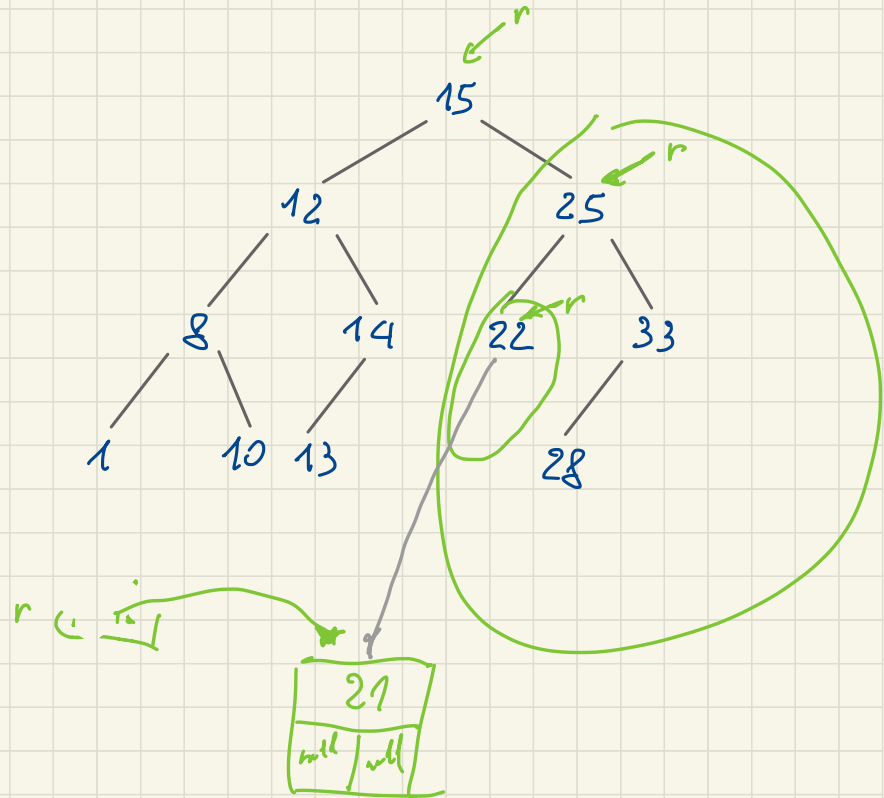
r.sx \leftarrow inserisci (r.sx, d)

ELSE

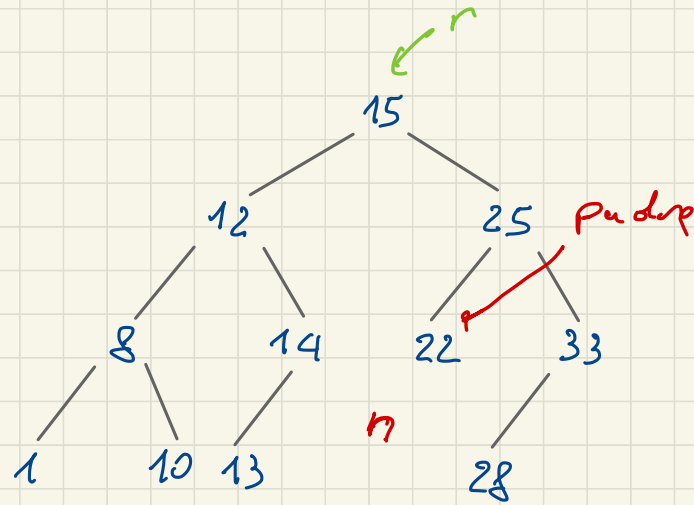
r.dx \leftarrow inserisci (r.dx, d)

RETURN r

k [21]



Inserimento iterativo



k [21]

FUNZIONE $inserisci(AlberoRicerca\ n, elemento\ d) \rightarrow AlberoRicerca$

$k \leftarrow d.chiave$

$t \leftarrow riferimento\ a\ nuovo\ nodo$

$t.chiave \leftarrow k$

$t.altriCampi \leftarrow d.altriCampi$

$t.sx \leftarrow null$

$t.dx \leftarrow null$

predefiniscono

nuovo nodo

$padre \leftarrow null$

$n \leftarrow r$

WHILE $n \neq null$ DO

$padre \leftarrow n$

 IF $k < n.chiave$ THEN $n \leftarrow n.sx$

 ELSE $n \leftarrow n.dx$

ricerca
posizione

IF $padre = null$ THEN

$r \leftarrow t$

Albero
nuovo

ELSE IF $k < p.chiave$ THEN

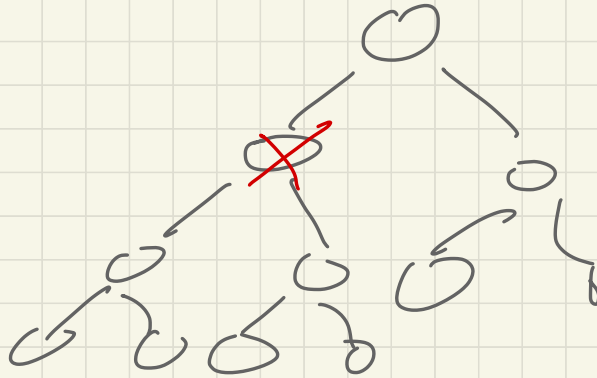
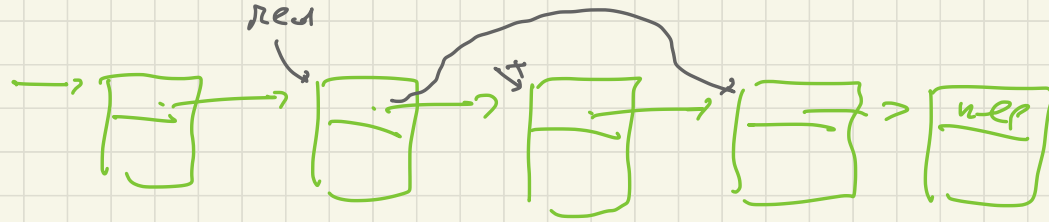
$padre.sx \leftarrow t$

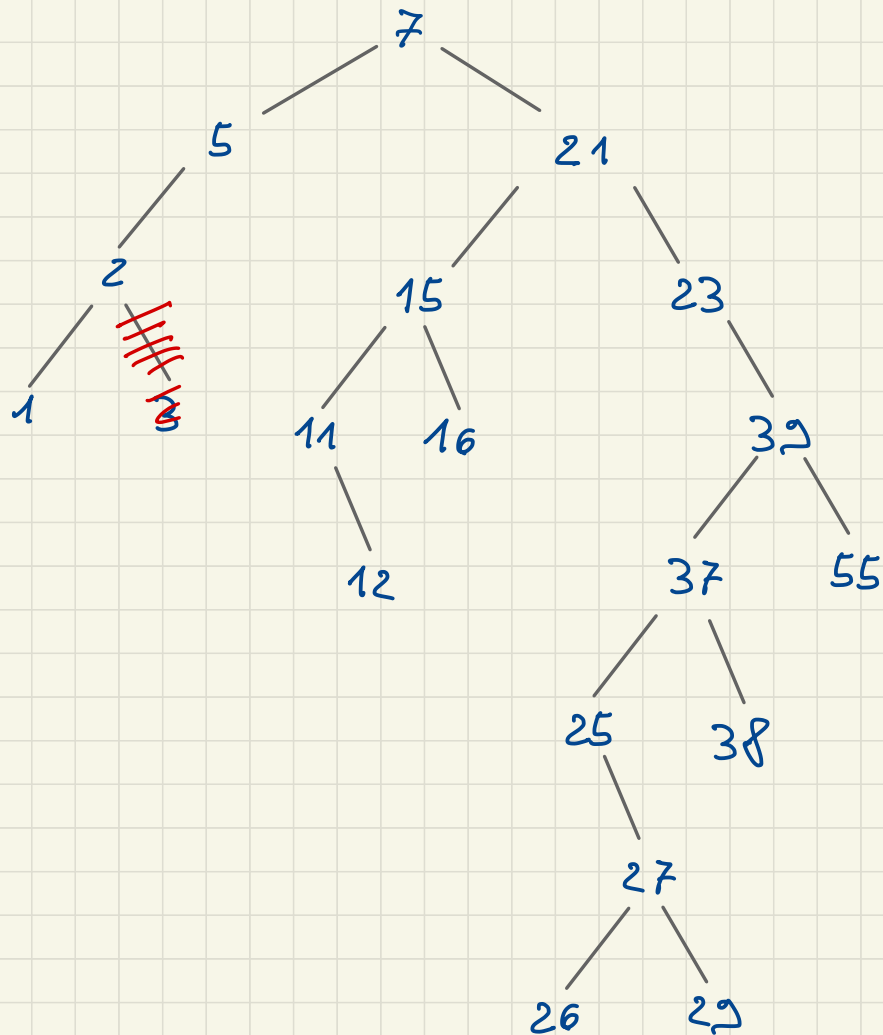
 ELSE $padre.dx \leftarrow t$

inserimento

RETURN r

Cancellation



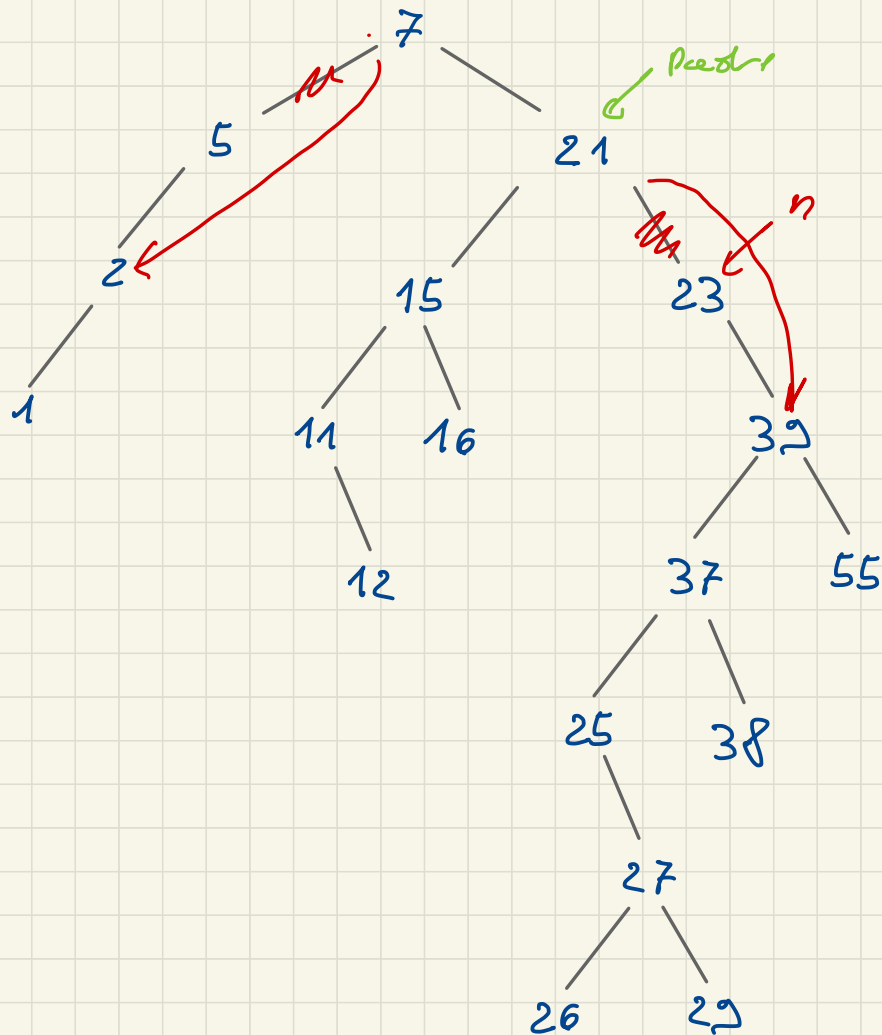


Cancellare 3

Cancellare il root figlio:

modificare perfettamente

nel nostro pattern



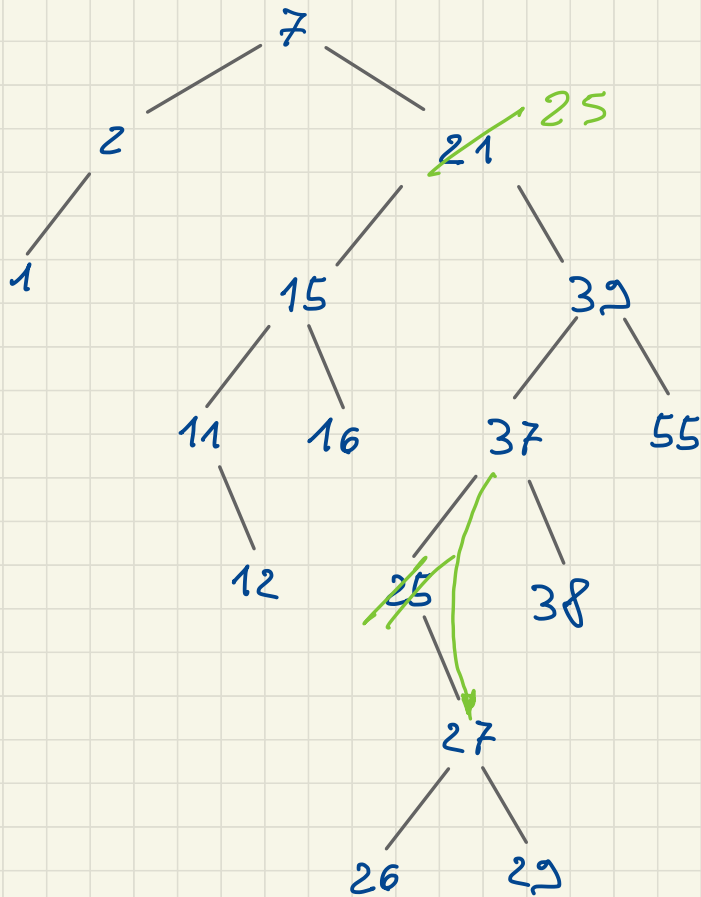
Cancella 23

Cancella 5

Il nodo da
cancellare ha
un unico figlio

↓
Modifico il padre

(caso particolare
radice)

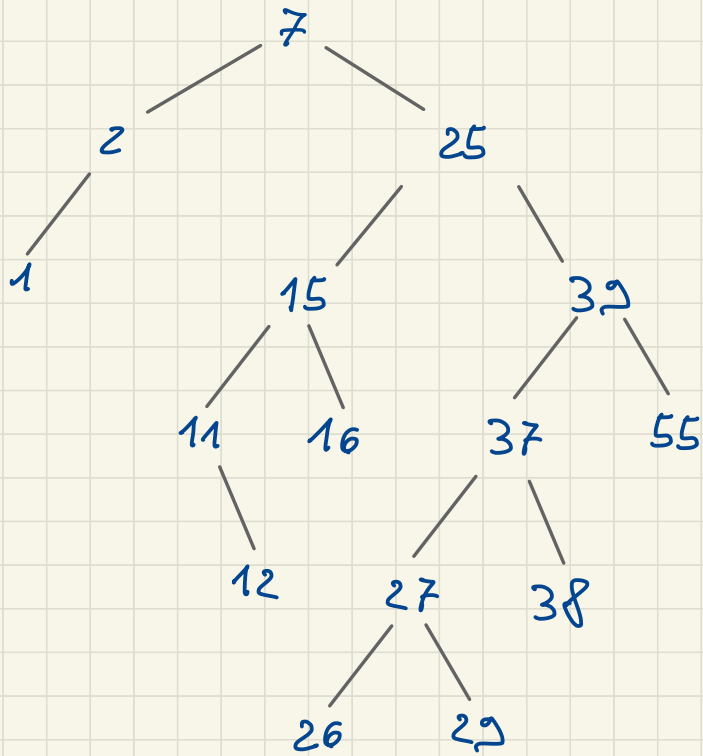


Cancella 21

node con estensi;
Right

- sostituisci il contenuto del nodo da cancellare con il contenuto del nodo di chiave più piccola del sottalbero dx del nodo da cancellare
- elimina il nodo che aveva chiave più piccola nel sottalbero dx

CASO PARTICOLARE: radice



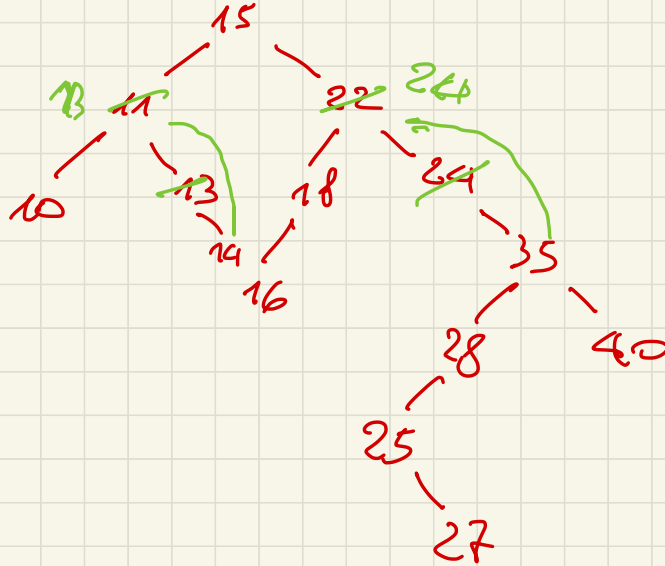
Esercizio

Disegnare l'albero di ricerca che si ottiene a partire da un albero vuoto inserendo uno dopo l'altro, nell'ordine indicato, i seguenti numeri:

15 11 13 22 24 18 35 18 25 16 40 10 27 14

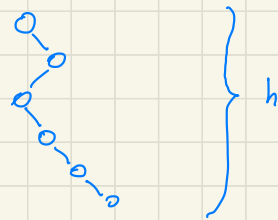
Disegnare come diventa l'albero se si rimuovono

22 11

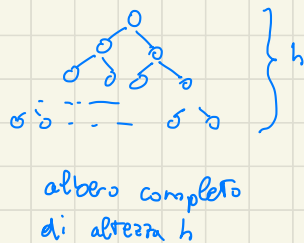


Alberi binari: n° nodi vs altezza

- Numero minimo di nodi per alberi di altezza h



- Numero massimo di nodi per alberi di altezza h



$$\Rightarrow h+1 \leq n \leq 2^{h+1}-1$$

$$\Rightarrow \lg_2(n+1) - 1 \leq h \leq n-1$$

1 5 10 12 15 20

1
 \
 5
 \
 10
 \
 12
 \
 15
 \
 20