AUMENTARE LE STRUTTURE DATI

- LE POTENZIALITA' DELLE STRUTTURE DATI FONDAMENTALI (ES. LISTE, TAVOLE HASH, ALBERI BINARI DI RICERCA, ECC.) POSSONO ESSERE "AUMENTATE" CON OPPORTUNE MODIFICHE ALLE STRUTTURE DATI STESSE, MEMORIZZANDO OPPORTUNE INFORMAZIONI AGGIUNTIVE (CHE DOVRANNO POI ESSERE MANTENUTE INSIEME ALLE ALTRE)

- ES. ALBERI PER STATISTICHE D'ORDINE

- UN ALBERO PER STATISTICHE D'ORDINE T E' UN ALBERO ROSSO-NERO CHE SUPPORTA ANCHE L'OPERAZIONE DI RICERCA DI UN ELEMENTO CON UN DATO RANGO

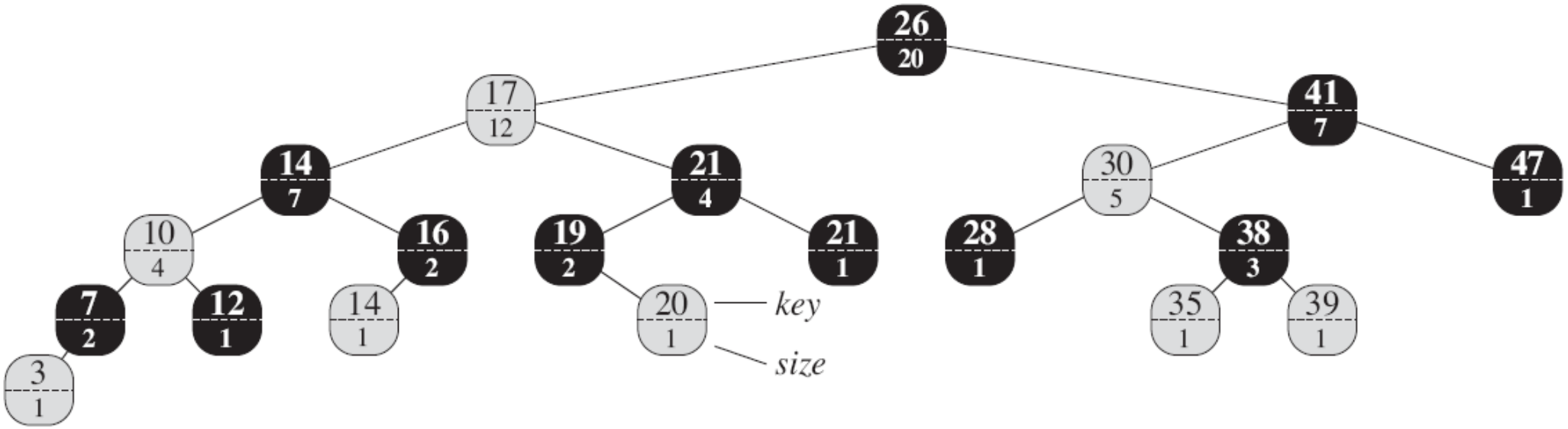
- IN AGGIUNTA AGLI ATTRIBUTI key, color, p, left, right, E' SUFFICIENTE MEMORIZZARE IN CIASCUN NODO x ANCHE UN ATTRIBUTO x.size CHE CONTIENE IL NUMERO DI NODI (INTERNI) NEL SOTTOALBERO RADICATO IN x (x COMPRESO)

- SI PONE T.nil.size = 0

- PER OGNI NODO INTERNO x VALE L'IDENTITA'

$$x.size = x.left.size + x.right.size + 1$$

ESEMPIO:



OS-SELECT(x, i)

```
1  r = x.left.size + 1
2  if i == r
3        return x
4  elseif i < r
5        return OS-SELECT(x.left, i)
6  else return OS-SELECT(x.right, i − r)
```

OS-SELECT$(T.root, i)$

COMPLESSITA': $O(\log n)$

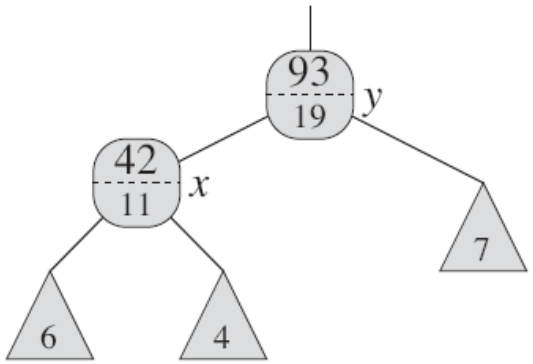
OS-RANK(T, x)

```
1  r = x.left.size + 1
2  y = x
3  while y ≠ T.root
4        if y == y.p.right
5              r = r + y.p.left.size + 1
6        y = y.p
7  return r
```

COMPLESSITA': $O(\log n)$

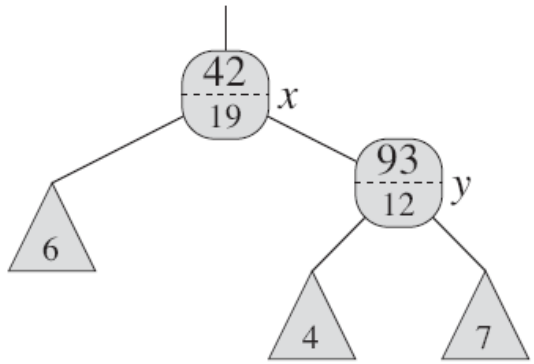
COME GESTIRE L'ATTRIBUTO size NEL CORSO DEGLI INSERIMENTI E CANCELLAZIONI ?

```
LEFT-ROTATE(T, x)
 1   y = x.right
 2   x.right = y.left
 3   if y.left ≠ T.nil
 4         y.left.p = x
 5   y.p = x.p
 6   if x.p == T.nil
 7         T.root = y
 8   elseif x == x.p.left
 9         x.p.left = y
10   else x.p.right = y
11   y.left = x
12   x.p = y
13   y.size = x.size
14   x.size = x.left.size + x.right.size + 1
```



LEFT-ROTATE(T, x)

RIGHT-ROTATE(T, y)



INSERIMENTO:

- DURANTE LA FASE DI DISCESA PER INNESTARE IL NUOVO NODO z SI INCREMENTA DI UN'UNITA' IL VALORE DELL'ATTRIBUTO size DEI NODI INCONTRATI
- LE MODIFICHE STRUTTURALI NEL CORSO DELLA SECONDA FASE DI RISALITA SONO PROVOCATE DALLE ROTAZIONI (AL MASSIMO DUE)
- LE RIGHE 13-14 IN LEFT-ROTATE (E ANALOGAMENTE IN RIGHT-ROTATE) SI FANNO CARICO DI AGGIORNARE CORRETTAMENTE I DUE ATTRIBUTI size INVALIDATI

CANCELLAZIONE:

- SI SEGUE IL CAMMINO SEMPLICE DALLA POSIZIONE ORIGINALE DEL NODO y FINO ALLA RADICE, DECREMENTANDO DI UN'UNITÀ IL VALORE DELL'ATTRIBUTO size DEI NODI INCONTRATI

- LE MODIFICHE STRUTTURALI NEL CORSO DELLA SECONDA FASE DI RISALITA SONO PROVOCATE DALLE ROTAZIONI (AL MASSIMO TRE)

- ANCHE IN QUESTO CASO, LE RIGHE 13-14 IN LEFT-ROTATE (E ANALOGAMENTE IN RIGHT-ROTATE) SI FANNO CARICO DI AGGIORNARE CORRETTAMENTE I DUE ATTRIBUTI size INVALIDATI

TEOREMA (AUMENTO DI UN ALBERO ROSSO-NERO)

- SIA f UN ATTRIBUTO CHE AUMENTA UN ALBERO ROSSO-NERO T DI n NODI.
- SUPPONIAMO CHE PER OGNI DATO NODO x IL VALORE $x.f$ SIA CALCOLABILE DALLE INFORMAZIONI RESIDENTI NEI NODI: x, $x.left$, $x.right$
- ALLORA E' POSSIBILE MANTENERE I VALORI DI f DURANTE LE OPERAZIONI DI INSERIMENTO E DI CANCELLAZIONE SENZA INFLUIRE ASINTOTICAMENTE SULLE PRESTAZIONI $O(\lg n)$.

ESERC121

14.1-5

Given an element x in an n-node order-statistic tree and a natural number i, how can we determine the ith successor of x in the linear order of the tree in $O(\lg n)$ time?

14.2-1

Show, by adding pointers to the nodes, how to support each of the dynamic-set queries MINIMUM, MAXIMUM, SUCCESSOR, and PREDECESSOR in $O(1)$ worst-case time on an augmented order-statistic tree. The asymptotic performance of other operations on order-statistic trees should not be affected.