Esercizi su ABR

Sostituzione di una chiave con un’altra
creazione di un ABR degenere a sinistra con le chiavi di un ABR qualunque.

Sostituzione di una chiave

Dato un albero binario di ricerca T, si descriva un algoritmo in cui una chiave x viene sostituita da una chiave y.
Si scriva un algoritmo per ottenere il risultato in O(h), dove h è l' altezza dell' albero.

Sostituzione di una chiave

Dato un albero binario di ricerca T, si descriva un algoritmo in cui una chiave x viene sostituita da una chiave y.
Si scriva un algoritmo per ottenere il risultato in O(h), dove h è l' altezza dell' albero.

Si usa delete per cancellare x e poi insert per inserire y.

substitute(T,x,y)

input: T è un puntatore alla radice di albero binario

prec: T è un ABR

output: il puntatore a un ABR ottenuto da T cancellando x e inserendo y

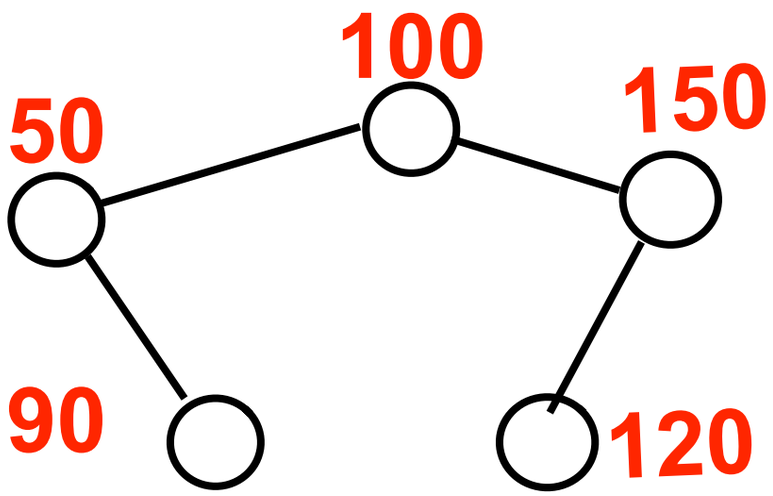
delete(T,x)

insert(T,y)

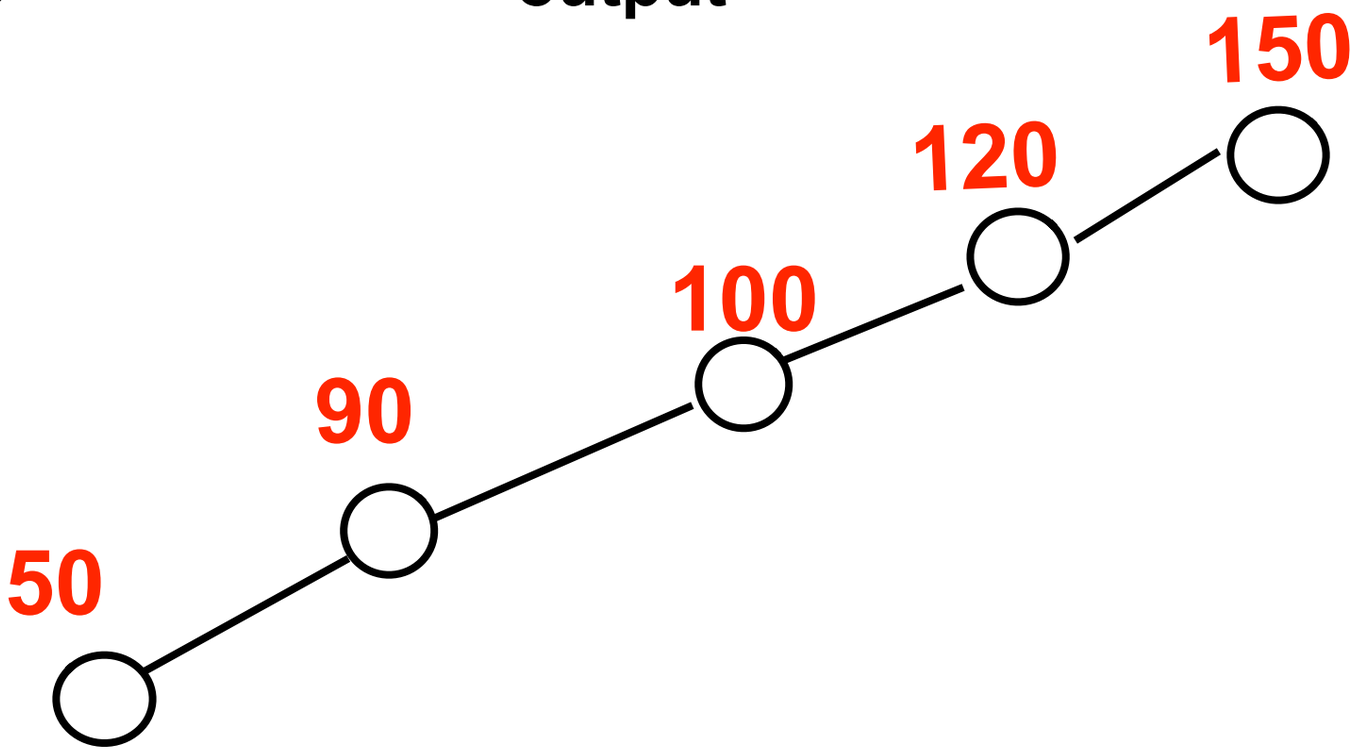
return

Creare un albero degenere a sinistra con le chiavi di un ABR qualunque

input, che non viene modificato

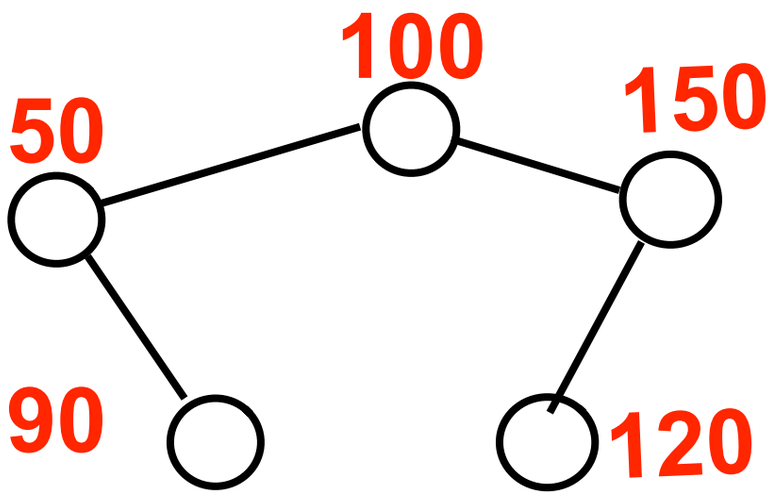


output

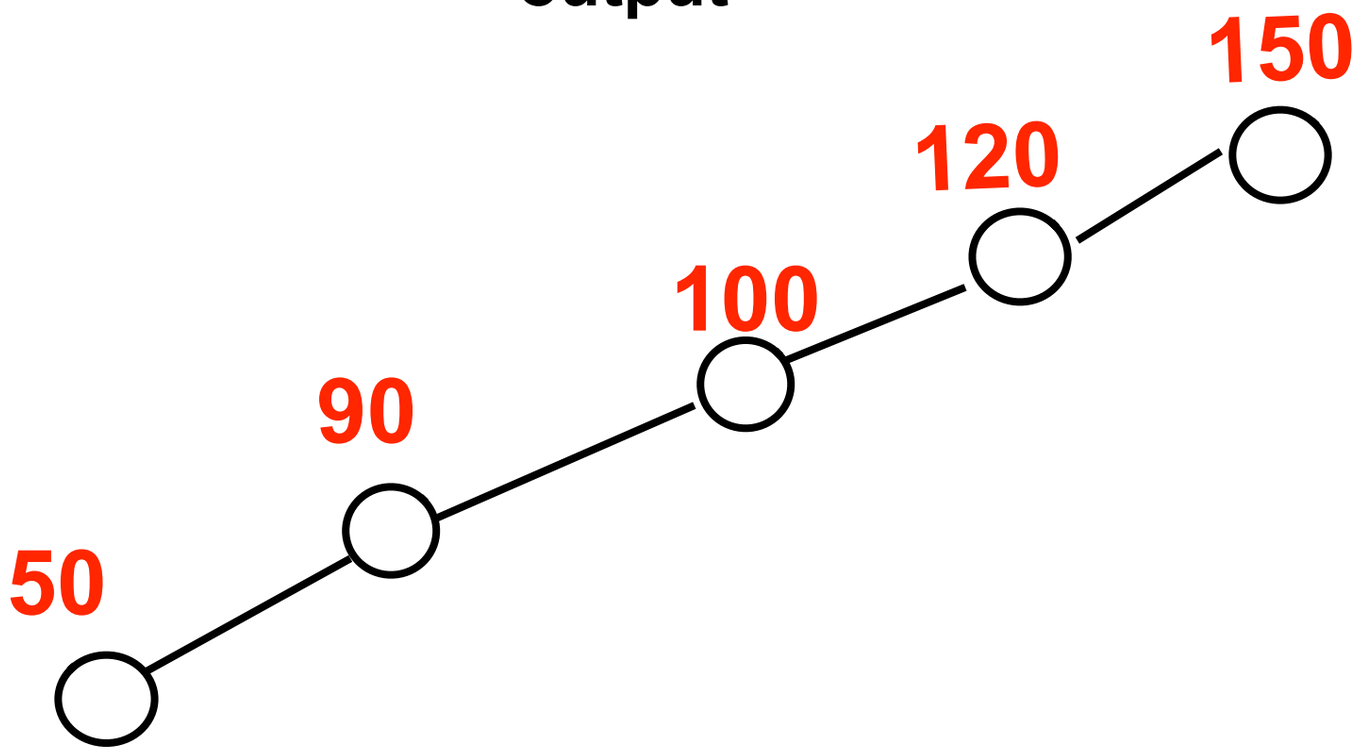


Creare un albero degenere a sinistra con le chiavi di un ABR qualunque

input, che non viene modificato



output



Si potrebbe fare una visita inorder rovesciata, partendo dal massimo e calcolando il predecessore di ogni nodo fino al minimo.

Ogni volta che viene visitato un nodo, si crea un nuovo nodo che avrà il prossimo visitato come figlio sinistro.

Pseudocodice

```
DegenereSin(T)
input: T è un puntatore alla radice di albero binario
prec: T è un ABR
output: il puntatore a un ABR degenere a sinistra con le stesse chiavi di T
if T == NIL return T
max = MAXIMUM(T) max è il nodo di chiave massima in T
Q = New(max.key)  new crea un nuovo nodo  puntato da Q con chiave max
e puntatori ai figli e al padre nil, Q è la radice dell'albero in output
z = Q
y = PREDECESSOR(max)
while y ≠ NIL do
y è il nodo che precede in T l'ultimo inserito in Q
    t = NEW(y.key)
    z.left = t;
    t.p=z
    z=t
    y=PREDECESSOR(y)
return Q
```