

```
// concatena due alberi di ricerca binaria
```

```
concatenate(TREE  $T_1$ , TREE  $T_2$ )
```

```
┌   TREE  $v = \max(T_1)$   
└   link( $v$ ,  $T_2$ ,  $T_2.value$ )
```

```
// collega un nodo padre  $p$  ad un nodo figlio  $u$ 
```

```
link(TREE  $p$ , TREE  $u$ , ITEM  $x$ )
```

```
┌   se  $u \neq \text{nil}$  allora  
└       // il nodo è stato cancellato  
         $u.parent = p$ 
```

```
┌   se  $p \neq \text{nil}$  allora  
└       se  $x < p.key$  allora  
             $p.left = u$   
        altrimenti  
             $p.right = u$ 
```

Si cerca il massimo valore contenuto in  $T_1$  e lo si “attacca” alla radice di  $T_2$  come figlio destro di  $v$ . concatenate ha costo  $\mathcal{O}(h)$ , dove  $h$  è l’altezza massima fra i due alberi. link ha costo  $\mathcal{O}(1)$ .