

Alberi Binari di Ricerca

Pseudocodice*

Algoritmo 1 TreeSearch(x, k)

TreeSearch(x, k)

```
1: while ( $x \neq \text{NIL}$ ) and ( $\text{key}[x] \neq k$ ) do
2:   if  $k < \text{key}[x]$  then
3:      $x \leftarrow \text{left}[x]$ 
4:   else
5:      $x \leftarrow \text{right}[x]$ 
6:   end if
7: end while
8: return  $x$ 
```

*T. H. Cormen, C. E. Leiserson, R. L. Rivest e C. Stein. *Introduction to Algorithms*. MIT Press, Second Edition, 2001.

Algoritmo 2 TreeInsert(T, z)

TreeInsert(T, z)

```
1: //  $p[z] = left[z] = right[z] = \text{NIL}$ 
2:  $y \leftarrow \text{NIL}$ 
3:  $x \leftarrow \text{root}(T)$ 
4: while  $x \neq \text{NIL}$  do
5:    $y \leftarrow x$ 
6:   if  $key[z] < key[x]$  then
7:      $x \leftarrow left[x]$ 
8:   else
9:      $x \leftarrow right[x]$ 
10:  end if
11: end while
12:  $p[z] \leftarrow y$ 
13: if  $y = \text{NIL}$  then
14:    $\text{root}(T) \leftarrow z$ 
15: else
16:   if  $key[z] < key[y]$  then
17:      $left[y] \leftarrow z$ 
18:   else
19:      $right[y] \leftarrow z$ 
20:   end if
21: end if
```

Algoritmo 3 TreeMin(x)

TreeMin(x)

```
1: if  $x = \text{NIL}$  then
2:   return  $\text{NIL}$ 
3: end if
4: while  $left[x] \neq \text{NIL}$  do
5:    $x \leftarrow left[x]$ 
6: end while
7: return  $x$ 
```

Algoritmo 4 TreeMax(x)

TreeMax(x)

```
1: if  $x = \text{NIL}$  then
2:   return  $\text{NIL}$ 
3: end if
4: while  $right[x] \neq \text{NIL}$  do
5:    $x \leftarrow right[x]$ 
6: end while
7: return  $x$ 
```

Algoritmo 5 TreeSuccessor(x)

TreeSuccessor(x)

```
1: if  $right[x] \neq \text{NIL}$  then
2:   return  $TreeMin(right[x])$ 
3: end if
4:  $y \leftarrow p[x]$ 
5: while ( $y \neq \text{NIL}$ ) and ( $x = right[y]$ ) do
6:    $x \leftarrow y$ 
7:    $y \leftarrow p[y]$ 
8: end while
9: return  $y$ 
```

Algoritmo 6 TreePredecessor(x)

TreePredecessor(x)

```
1: if  $left[x] \neq \text{NIL}$  then
2:   return  $TreeMax(left[x])$ 
3: end if
4:  $y \leftarrow p[x]$ 
5: while ( $y \neq \text{NIL}$ ) and ( $x = left[y]$ ) do
6:    $x \leftarrow y$ 
7:    $y \leftarrow p[y]$ 
8: end while
9: return  $y$ 
```

Algoritmo 7 TreeDelete(T, z)

TreeDelete(T, z)

```
1: if ( $left[z] = \text{NIL}$ ) or ( $right[z] = \text{NIL}$ ) then
2:    $y \leftarrow z$ 
3: else
4:    $y \leftarrow \text{TreeSuccessor}(z)$ 
5: end if
6: if  $left[y] \neq \text{NIL}$  then
7:    $x \leftarrow left[y]$ 
8: else
9:    $x \leftarrow right[y]$ 
10: end if
11: if  $x \neq \text{NIL}$  then
12:    $p[x] \leftarrow p[y]$ 
13: end if
14: if  $p[y] = \text{NIL}$  then
15:    $root(T) \leftarrow x$ 
16: else
17:   if  $y = left[p[y]]$  then
18:      $left[p[y]] \leftarrow x$ 
19:   else
20:      $right[p[y]] \leftarrow x$ 
21:   end if
22: end if
23: if  $y \neq z$  then
24:    $key[z] \leftarrow key[y]$ 
25:   // copia i dati satellite di  $y$  in  $z$ 
26: end if
27: return  $y$ 
```
