Corso di Algoritmi e Strutture Dati
Anno Accademico 2017–2018
Prof. Giovanni Pighizzini

Pseudocodice

E. Alberi binari di ricerca

E.1. Ricerca (versione ricorsiva)

Funzione *trova (AlberoRicerca r, intero x)* $\rightarrow$ *Nodo*
  if $r = null$ **then return** $null$
  else if $x < r.dato$ **then return** $trova(r.sx, x)$
  else if $x > r.dato$ **then return** $trova(r.dx, x)$
  else return r

E.2. Ricerca (versione iterativa)

Funzione *trova (AlberoRicerca r, intero x)* $\rightarrow$ *Nodo*
  $n \leftarrow r$
  while $n \neq null$ **and** $n.dato \neq x$ **do**
    if $x < n.dato$ **then** $n \leftarrow n.sx$
    else $n \leftarrow n.dx$
  return n

E.3. Inserimento (versione ricorsiva)

Procedura *inserisci (AlberoRicerca r, intero x)*
  `/* Inserisce nell'albero di ricerca riferito dal parametro` r `un nuovo nodo`
     `contenente il valore di` x`, se non è già presente.`
     `Il parametro` r`, contenente il riferimento alla radice, deve essere passato`
     `per riferimento.` `*/`
  if $r = null$ **then**
    $t \leftarrow$ riferimento a nuovo nodo
    $t.dato \leftarrow x$
    $t.sx \leftarrow null$
    $t.dx \leftarrow null$
    $r \leftarrow t$
  else if $x < r.dato$ **then** $inserisci(r.sx, x)$
  else if $x > r.dato$ **then** $inserisci(r.dx, x)$

E.4. Inserimento (versione iterativa)

```
Procedura inserisci (AlberoRicerca r, intero x)
  /* Inserisce nell'albero di ricerca riferito dal parametro r un nuovo nodo
     contenente il valore di x, se non è già presente.
     Il parametro r, contenente il riferimento alla radice, deve essere passato
     per riferimento.                                                          */

  // Preparazione del nodo da inserire
  t ← riferimento a nuovo nodo
  t.dato ← x
  t.sx ← null
  t.dx ← null

  // Ricerca la posizione dove inserire
  padre ← null
  n ← r
  while n ≠ null do
      padre ← n
      if x < n.dato then n ← n.sx
      else n ← n.dx
  // in questo punto padre si riferisce al nodo sotto il quale inserire
  // (contiene null se l'inserimento va fatto alla radice - albero vuoto)

  // Inserisci
  if padre = null then r ← t
  else if x < padre.dato then padre.sx ← t
  else if x > padre.dato then padre.dx ← t
```

E.5. Cancellazione

```
Procedura cancella (AlberoRicerca r, intero x)
  /* Elimina dall'albero di ricerca il nodo contenente il valore di x, se non
     è già presente.
     Il parametro r, contenente il riferimento alla radice, deve essere passato
     per riferimento.                                                          */

  // Ricerca il nodo da cancellare e il suo nodo padre
  padre ← null
  n ← r
  while n ≠ null and n.dato ≠ x do
      padre ← n
      if x < n.dato then n ← n.sx
      else n ← n.dx

  // Cancella il nodo se è stato trovato
  if n ≠ null then
      if n.sx = null then                    // manca figlio sinistro:  sostituisci il
          if padre ≠ null then                   // nodo con il suo sottoalbero destro
              if n.dato < padre.dato then
                  padre.sx ← n.dx
              else
                  padre.dx ← n.dx
          else
              r ← r.dx
      else if n.dx = null then               // manca figlio destro:  sostituisci il
          if padre ≠ null then               // nodo con il suo sottoalbero sinistro
              if n.dato < padre.dato then
                  padre.sx ← n.sx
              else
                  padre.dx ← n.sx
          else
              r ← r.sx
      else                                                  // ci sono entrambi i figli
          // ricerca il nodo contenente la chiave più grande
          // del sottoalbero sinistro
          t ← n
          m ← n.sx
          while m.dx ≠ null do
              t ← m
              m ← m.dx
          // in questo punto m si riferisce al nodo contenente la chiave più
          // grande del sottoalbero sinistro, t al padre di m
          n.dato ← m.dato                // copia i dati presenti in m nel nodo n
          // elimina il nodo riferito da m
          if t = n then                 // se il nodo m è figlio sinistro del nodo n
              n.sx ← m.sx
          else
              t.dx ← m.sx
```