Algorithm Design

M. T. Goodrich and R. Tamassia

John Wiley & Sons

Solution of Exercise R-2.7

We assume that a vector S is used for the representation of the tree T. Also, we assume that S contains as elements positions. We give the pseudo-code for the various methods:

Algorithm root():
  return S.elemAtRank(1)

Algorithm parent(v):
  return S.elemAtRank($\lfloor p(v)/2 \rfloor$)

Algorithm leftchild(v):
  return S.elemAtRank($2p(v)$)

Algorithm rightchild(v):
  return S.elemAtRank($2p(v)+1$)

Algorithm sibling(v):
  if T.isRoot(v) **then**
    throw InvalidNodeException
  if $p(v)$ is even **then**
    return S.elemAtRank($p(v)+1$)
  else
    return S.elemAtRank($p(v)-1$)

Algorithm isInternal(v):
  return ($2p(v)+1 <$ S.size()-1) $\wedge$ (S.elemAtRank($2p(v)$)$\neq$ NULL)

Algorithm isExternal(v):
  return $\neg$ T.isInternal(v)

Algorithm isRoot(v):
  return (v=S.elemAtRank(1))

Algorithm size():
  $c \leftarrow 0$
  for $i \leftarrow 1, 2, \ldots, S.\mathsf{size}() - 1$ **do**
    if $S.\mathsf{elemAtRank}(i) \neq$ NULL **then**
      $c \leftarrow c + 1$
  return c