C-6.23

```
Algorithm eulerTour(Tree $T$, Position $v$):
  $state \leftarrow start$
  while $state \neq done$ do
    if $state = start$ then
      if $T$.isExternal($v$) then
        left action; below action; right action
        $state \leftarrow = done$
      else
        left action
        $state \leftarrow$ on_the_left
        $v \leftarrow v.leftchild$
    if $state =$ on_the_left then
      if $T$.isExternal($v$) then
        left action; below action; right action
        $state =$ from_the_left
        $v \leftarrow v.parent$
      else
        left action
        $v \leftarrow v.leftchild$
    if $state =$ from_the_left then
      below action
      $state \leftarrow$ on_the_right
      $v \leftarrow v.right$
    if $state =$ on_the_right then
      if $T$.isExternal($v$) then
        $state =$ from_the_right
        left action; below action; right action
        $v \leftarrow v.parent$
      else
        left action
        $state \leftarrow$ on_the_left
        $v \leftarrow v.left$
    if $state =$ from_the_right then
      right action
      if $T$.isRoot($v$) then
        $state \leftarrow done$
      else
        if $v$ is left child of parent then
          $state \leftarrow$ from_the_left
        else
          $state \leftarrow$ from_the_right
        $v \leftarrow v.parent$
```