

function AVLTreeDeleteMin(*locative T*) **returns** (**key, info, boolean**)

comment: In the tree pointed to by *T*, deletes the minimum key and returns that record's Key and Info fields, and also returns true if and only if the deletion caused the height of the tree to decrease;

```
begin
  declare K: key;
  declare I: info;
  declare decreased: boolean;
  declare B: integer;
  if T =  $\Lambda$  then error;
  if LC(T) =  $\Lambda$ 
    then begin
      K  $\leftarrow$  Key(T);
      I  $\leftarrow$  Info(T);
      T  $\leftarrow$  RC(T);
      return (K, I, true)
    end
  else begin
    (K, I, decreased)  $\leftarrow$  AVLTreeDeleteMin(LC(T));
    if not decreased
      then return (K, I, false);
      else case Balance(T) of
        -1: Balance(T)  $\leftarrow$  0;
           return (K, I, true);
        0: Balance(T)  $\leftarrow$  +1;
           return (K, I, false);
        +1: case Balance(RC(T)) of
          +1: Rotate(T, -1);
              Balance(T)  $\leftarrow$  0;
              Balance(LC(T))  $\leftarrow$  0;
              return (K, I, true);
          0: Rotate(T, -1);
              Balance(T)  $\leftarrow$  -1;
              Balance(LC(T))  $\leftarrow$  +1;
              return (K, I, false);
          -1: B  $\leftarrow$  Balance(LC(RC(T)));
              Rotate(RC(T), +1);
              Rotate(T, -1);
              Balance(T)  $\leftarrow$  0;
              if B = +1
                then Balance(LC(T))  $\leftarrow$  -1
                else Balance(LC(T))  $\leftarrow$  0;
              if B = -1
                then Balance(RC(T))  $\leftarrow$  +1
                else Balance(RC(T))  $\leftarrow$  0;
              return (K, I, true);
        end case
      end case
    end
  end
end .
```