```
from binary_search_tree import TreeMap

class SplayTreeMap(TreeMap):
  """Sorted map implementation using a splay tree."""

  #--------------------------------- splay operation ---------------------------
-----
  def _splay(self, p):
    while p != self.root():
      parent = self.parent(p)
      grand = self.parent(parent)
      if grand is None:
        # zig case
        self._rotate(p)
      elif (parent == self.left(grand)) == (p == self.left(parent)):
        # zig-zig case
        self._rotate(parent)             # move PARENT up
        self._rotate(p)                  # then move p up
      else:
        # zig-zag case
        self._rotate(p)                  # move p up
        self._rotate(p)                  # move p up again

  #---------------------------- override balancing hooks -----------------------
-----
  def _rebalance_insert(self, p):
    self._splay(p)

  def _rebalance_delete(self, p):
    if p is not None:
      self._splay(p)

  def _rebalance_access(self, p):
    self._splay(p)
```