```
PriorityQueue (>:E,E→Bool, E, C): trait
  % Enumerate by order on elements
  assumes TotalOrder (E for T)
  includes Integer
  introduces
    empty: → C
    add: E, C → C
    count: E, C → Int
    __ ∈ __: E, C → Bool
    head: C → E
    tail: C → C
    len: C → Int
    isEmpty: C → Bool
  asserts
    C generated by empty, add
    C partitioned by head, tail, isEmpty
    ∀ e, e1: E, q: C
      count(e, empty) == 0;
      count(e, add(e1, q)) ==
        count(e, q) + (if e = e1 then 1 else 0);
      e ∈ q == count(e, q) > 0;
      head(add(e, q)) ==
        if q = empty ∨ e > head(q) then e
        else head(q);
      tail(add(e, q)) ==
        if q = empty ∨ e > head(q) then q
        else add(e, tail(q));
      len(empty) == 0;
      len(add(e, q)) == len(q) + 1;
      isEmpty(q) == q = empty
  implies
    Container (add for insert)
    ∀ e, e1, e2: E, q: C
      add(e1, add(e2, q)) = add(e2, add(e1, q));
      len(q) ≥ 0;
      add(e, q) ≠ empty
    converts count, ∈, head, tail, len, isEmpty
      exempting head(empty), tail(empty)
```