PISTAS.

1. Pista 1: Recuerda que los elementos en un std::set se almacenan en orden ascendente. Si te estás preguntando por qué un elemento en particular está en una posición determinada, piensa en cómo se comparan los elementos en términos de ordenamiento.
2. Pista 2: Si estás teniendo problemas para insertar un elemento en un std::set, asegúrate de que estás usando el método **insert()** y no el método **emplace()** o el método **push\_back()**, que no son válidos para std::set.
3. Pista 3: Si estás intentando eliminar un elemento de un std::set, asegúrate de que estás utilizando el método **erase()** y que estás pasando el elemento correcto como argumento.
4. Pista 4: Si estás teniendo dificultades para recorrer los elementos de un std::set con un loop for-each, recuerda que debes declarar la variable de iteración como una referencia constante para evitar copias innecesarias de los elementos.
5. Pista 5: Si estás interesado en explorar más a fondo las propiedades de un std::set, investiga los métodos que están disponibles para manipular conjuntos, como **find()**, **count()**, **lower\_bound()**, y **upper\_bound()**. Estos métodos pueden ser útiles para resolver problemas específicos con conjuntos.