在实践中，重要的是定义了“正常行为”的<运算符的类型可以用作键。

如果键类型本质上是无序的，或者如果性能测试发现散列可能解决的问题，则使用无序容器。

**2. 红黑树的规则**

红黑树必须满足以下性质：

1. **节点颜色**：
   * 每个节点是红色或黑色。
2. **根节点**：
   * 根节点必须是黑色。
3. **叶子节点（NIL节点）**：
   * 所有叶子节点（NIL节点，表示空指针）都是黑色。
4. **红色节点的子节点**：
   * 如果一个节点是红色，则它的两个子节点必须是黑色（即不能有两个连续的红色节点）。
5. **黑高平衡**：
   * 从任一节点到其每个叶子节点的所有路径上，黑色节点的数量必须相同（称为黑高）。

**3. 红黑树的特点**

**(1) 自平衡**

* 红黑树通过颜色规则和旋转操作（左旋和右旋）保持平衡。
* 插入和删除操作可能会破坏红黑树的规则，但通过调整颜色和旋转可以恢复平衡。

**(2) 高效操作**

* **查找**：时间复杂度为 O(log⁡N)*O*(log*N*)，因为树的高度是平衡的。
* **插入**：时间复杂度为 O(log⁡N)*O*(log*N*)，插入后可能需要调整颜色和旋转。
* **删除**：时间复杂度为 O(log⁡N)*O*(log*N*)，删除后可能需要调整颜色和旋转。

**(3) 有序性**

* 红黑树是一种二叉搜索树，因此中序遍历可以得到有序的序列。

**(4) 空间复杂度**

* 每个节点需要额外的空间存储颜色信息（通常是一个布尔值或枚举类型）。

11.4