## Trie树

.字典树，一种专门用于字符串匹配的数据结构。

.Trie树的构建过程：  
如果有6个字符串，how 、hi 、 hello so see 。我们希望在里面多次查找某个字符串是否存在。根节点不包含任何信息，每个节点表示一个字符串中的字符，从根节点到红色节点的一条路径表示一个字符串。

当我们在Trie树中查找一个字符串的时候，比如查找her,将我们要查找的字符串分割成单个的字符h,e,r，然后从Trie树的根节点出发开始匹配。

.Trie树的实现过程

主要操作有两个，一个是将字符串集合构造成为Trie树，这个过程就是讲字符串插入到Trie树的过程，另外一个是在Trie树种查询一个字符串。

Trie树是一个多叉树，多叉树怎么存储一个节点的所有子节点的指针呢？经典的存储方式，是使用散列表的思想，假设我们的字符串只有26个小写字母，我们在数组中下标为0的位置，存储指向子节点a的指针，下标为1的位置存储指向节点b的指针，以此类推，下标为25的位置，存储的是指向节点z的指针，如果某个字符的子节点不存在，我们就在对应的下标的位置存储null。

.Trie树的代码实现

## 并查集

组团、配对问题

熟练并查集的模板

## AVL树和红黑树的实现和特性

AVL树

平衡因子：取值范围：-1、0、1

通过旋转操作来保持平衡

红黑树