

Trie树



主讲人：邓哲也



大纲

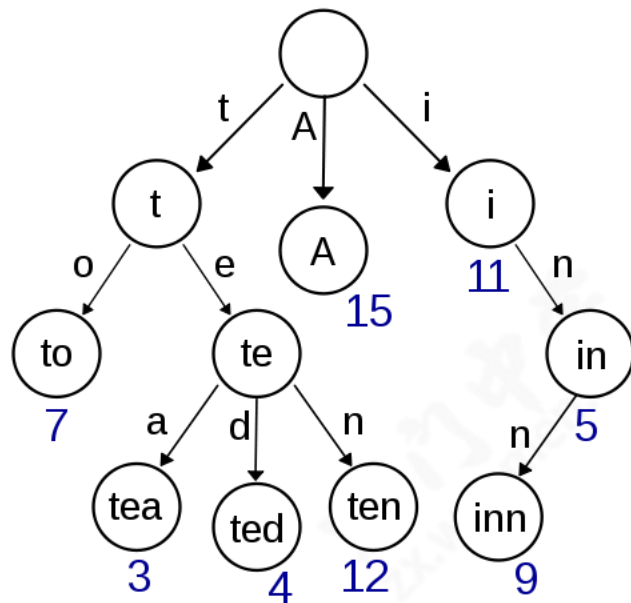
➤ Trie的定义

➤ 代码实现

➤ Trie的应用

Trie的定义

- trie, 又称前缀树或字典树, 是一种有序树, 用于保存关联数组, 其中的键通常是字符串。
- 与二叉查找树不同, 键不是直接保存在节点中, 而是由节点在树中的位置决定。
- 一个节点的所有子孙都有相同的前缀, 也就是这个节点对应的字符串, 而根节点对应空字符串。
- 一般情况下, 不是所有的节点都有对应的值, 只有叶子节点和部分内部节点所对应的键才有相关的值。



代码实现

- 假设trie树里存的都是小写字母字符串。
 - `go[i]`表示沿着字母i会走到的下一个节点
 - `end`表示当前节点是否是一个字符串的最后一个字符。
-
- **`struct node{`**
 - **`int go[26], end;`**
 - **`}t[N];`**
 - **`int root = 1, tot = 1;`**

代码实现

- 插入一个字符串 `str[]`
- `void insert(char *str) {`
- `int x = root;`
- `for (int i = 0; str[i]; i++) {`
- `int k = str[i] - 'a';`
- `if (!t[x].go[k])`
- `t[x].go[k] = ++ tot;`
- `x = t[x].go[k];`
- `}`
- `t[x].end = 1;`
- `}`

代码实现

- 查找一个字符串 `str[]`
- ```
int search(char *str) {
 int x = root;
 for (int i = 0; str[i]; i++) {
 int k = str[i] - 'a';
 if (!t[x].go[k]) return 0;
 x = t[x].go[k];
 }
 return t[x].end;
}
```

# Trie的应用

- 给字符串排序
- 统计以某个字符串作为前缀的单词数量

下节课再见