

检索结构树（Trie 树）的定义

```
enum TypeofNode {BranchNode,KeyNode}; //Trie 树结点类型：分支结点和关键词结点
const int M = 27;
/*Trie 树结点类定义*/
class TrieNode
{
public:
    TypeofNode NodeType;    //指明该结点一个关键词(叶结点)还是分支结点
    union
    {
        TrieNode* link[M]; //分支结点:指向各子树的指针
        char *key;          //关键词结点:关键词
    };
};
/*Trie 树类定义*/
class Trie
{
public:
    Trie(TrieNode* rt){root=rt;}
    TrieNode* TrieSearch( char word[] ); // word 为待查找的字符串，字符串尾包含结束符
    //int insert(...);                  //其他操作
    //int remove(...);
private:
    TrieNode* root;
};
算法 T（检索结构查找）
//在 M 叉 Trie 树中查找单词 word[]
TrieNode* Trie::TrieSearch( char word[] )
{
    TrieNode* p=root;
    TrieNode* x;
    int i=0,pos;
    while(true)
    {
        if(word[i]>='A' && word[i]<='Z') pos=word[i]-'A'+1;
        else if(word[i]>='a' && word[i]<='z') pos=word[i]-'a'+1;
        else pos=0;    //字符串结束符
        x=p->link[pos];
        if(x==NULL) return NULL;
        if(x->NodeType==BranchNode)    //此时 x 为分支结点
            p=x;
        else break;
        i++;
    }
    //此时 x 为关键词结点
    if(strcmp(x->key,word)==0) return x;
    else return NULL;
}
```