[**Trie树**](http://www.cppblog.com/guyuecanhui/articles/76924.html)

可以申请静态的结点数组，则Delete操作只要把数组的指针置为首地址/\*内\*/

http://www.cppblog.com/Images/OutliningIndicators/None.gif#define keyNum 26  
http://www.cppblog.com/Images/OutliningIndicators/None.gif#define maxSize 10000  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifstruct TrieNode{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    TrieNode \*link[keyNum];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int num[keyNum];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    bool iselement[keyNum];  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    void init(){  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        memset(link,0,sizeof(link));  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        memset(num,0,sizeof(num));  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        memset(iselement,0,sizeof(iselement));  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif};  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gif/\*\*//\*  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gifTrieNode tn[maxSize];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gifTrieNode \*cnt= tn;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif\*/  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifclass TrieTree{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gifpublic:  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    TrieNode\* root;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    TrieTree(){  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        root= 0;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    void Init(){  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        root= new TrieNode;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        /\*\*//\*root= cnt++;\*/  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        root->init();  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    bool Search( char \*x ){  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if( \*x==0 )    return false;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        TrieNode\* current= root;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        while( \*x ){  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            if( current->link[\*x-'a'] )  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                current= current->link[\*x-'a'];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            else break;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            x++;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if( current->iselement[\*x-'a'] || \*x==0 )  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            return true;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        else  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            return false;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    void Insert( char x[] ){  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        TrieNode \*current= root;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        int i= 1;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        while( x[i] ){  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif            if( current->link[x[i-1]-'a']==NULL ){  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                current->link[x[i-1]-'a']= new TrieNode;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif                /\*\*//\*current->link[x[i-1]-'a']= cnt++;\*/  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                (current->link[x[i-1]-'a'])->init();  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            current= current->link[x[i-1]-'a'];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            i++;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        (current->num[x[i-1]-'a'])++;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        current->iselement[x[i-1]-'a']= true;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    void Delete( TrieNode\* t ){  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        int i;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        for( i= 0; i<keyNum; i++ )  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            if( t->link[i] )    Delete(t->link[i]);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        memset( t->num,0,sizeof(t->num) );  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        memset( t->iselement,0,sizeof(t->iselement) );  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        delete(t);  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        /\*\*//\*cnt= tn;\*/  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif};