

05-E2

二叉树

先序遍历：迭代算法

凡是过往，皆为序章

我是如来最小之弟

邓俊辉

deng@tsinghua.edu.cn

# 序曲

```
template <typename T, typename VST> static void visitAlongVine
( BinNodePosi<T> x, VST & visit, Stack < BinNodePosi<T> > & S ) { //分摊 $O(1)$ 

    while ( x ) { //反复地

        visit( x->data ); //访问当前节点

        S.push( x->rc ); //右孩子（右子树）入栈（将来逆序出栈）

        x = x->lc; //沿藤下行

    } //只有右孩子、NULL可能入栈—增加判断以剔除后者，是否值得？

}
```

```
template <typename T, typename VST>

void travPre_I2( BinNodePosi<T> x, VST & visit ) {

    Stack < BinNodePosi<T> > S; //辅助栈

    while ( true ) { //以右子树为单位，逐批访问节点

        visitAlongVine( x, visit, S ); //访问子树x的藤蔓，各右子树（根）入栈缓冲

        if ( S.empty() ) break; //栈空即退出

        x = S.pop(); //弹出下一右子树（根）

    } // #pop = #push = #visit =  $O(n)$  = 分摊 $O(1)$ 

}
```

# 实例

