

5. 二叉树

先序遍历

算法B

邓俊辉

deng@tsinghua.edu.cn

```

❖ template <typename T, typename VST>

static void visitAlongLeftBranch

( 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; //沿左侧链下行

    } //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

visitAlongLeftBranch(x, visit, S); //访问子树x的左侧链, 右子树入栈缓冲

if (S.empty()) break; //栈空即退出

x = S.pop(); //弹出下一子树的根

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

实例

