2020.09.23：

1. 线索二叉树的应用是什么：由于线索二叉树只有终端结点的指针域才指向前驱和后继结点，那要想找到一个结点的前驱或者后继，岂不是这个结点限定为终端节点。
2. 线索二叉树修改成带头结点的线索二叉树，双向二叉链表理解一下，以及它的作用。
3. Typedef enum PointerTag{Link,Thread}的意思理解下
4. 二叉树的遍历：递归，非递归（利用栈实现、利用线索二叉树实现）代码梳理
5. 二叉树的先序中序后序的递归非递归代码梳理，特别是非递归的先序和后序的遍历代码
6. https://zhidao.baidu.com/question/1113376210070722899.html试说明是否存在这样的二叉树，可以实现后序线索树进行后序遍历时不使用栈？对前序线索二叉树进行前序遍历
7. 为什么线索二叉树可以不用二叉树，为什么三种遍历不用递归可以用栈实现。
8. 演示一下：先序和中序遍历的非递归流程，在纸上画画。
9. 为什么先序中序遍历时不需要三叉结点，而后序非递归遍历却需要？先序中序非递归还是没理解。

**遍历时后序时先把孩子全部遍历了再回到父母节点，而回寻父母节点的方法只能在定义的节点结构中添加一指针，指向该节点的父母节点。为什么先序中序不会遇到这个问题**

<https://blog.csdn.net/SZU_Crayon/article/details/81125122>

1. Char\* m=’helle’字符串指针是可以直接定义字符串，然后返回的就是指针吗？字符串的字符和串一个用双引号，一个用单引号。字符串遇到\0代表结束？
2. 结构体的内容是占用一片连续的存储空间。
3. Malloc函数分配的内存空间是一片连续的存储空间吗。
4. Int型占的字节。
5. gets\_s(指针)会报错，写数组的首字母就不会。
6. 形参用数组名或者指针，int a[] int \*p？？
7. 循环队列代码中，用malloc是进行堆分配？Malloc分配的内存是连续的？他更灵活为什么，结合字符串定长顺序代码看一下？字符串定长顺序存储结构是怎么写的代码？Exit(OVERFLOW)什么意思，怎么执行的？返回循环队列的长度，为什么形参不用引用（因为不需要修改Q的值）
8. 结构体SqQueue分配的空间是一块连续的存储空间，为什么要把rear和front与base放在一个结构体里，有什么好处？
9. 看下链队Status DeQueue(LinkQueue &Q, QElemType &e) //入队出队操作会更新rear或者front，所以Q要用引用 和循环队的出队入队用引用的函数。同时看下输出链队、循环队、链表的值要怎么做，看链栈顺序栈字符串三种顺序链式形式，也是用malloc分配内存返回首地址？（循环队列用malloc分配一串连续空间并返回数组的首地址，链队、链栈用malloc就是创建结点。顺序栈用malloc也是创建一片连续存储空间并返回base指针，也就是数组首地址） 链队链栈判断长度函数（顺序栈、循环队列用top-base、rear-front 链队、链栈用指针->next为空，并计数。或者形参里用icount变量。）、啥时候用引用（需要修改指针时）、if (S.top - S.base == S.stacksize)如果一个内存占据好几个直接，指针加1是直接加好几个字节吧？（是）
10. #define OVERFLOW -2返回值会是-2？
11. Malloc和new的区别，前者对应free后者对应delete

21、2020-10-12

明天把stack库函数看一下，查看能否可以把栈1赋值给栈2。其次，查看下其具体的函数，其次c++的stack<int>st的意思。同时一下代码报错的原因。

STL中的迭代器是什么（<https://www.cnblogs.com/curo0119/p/8012232.html>）

STL快速入门：http://c.biancheng.net/stl/

查看栈stack的库函数，如何把

#include<stack>

#include<queue>

#include<iostream>

#include<stdlib.h>

using namespace std;

void main()

{

stack<int> std1,std2;

int a[] = { 1,3,1,5 };

for (int i = 0; i <= 3; i++)

{

std2.push(a[i]);

}

for (int i = 0; i <= 3; i++)

{

cout<<std2.top()<<endl;

std2.pop();

}

std1 = std2;

cout << "std1中的元素为" << endl;

for (int i = 0; i <= 3; i++)

{

cout << std1.top() << endl;

std1.pop();

}

}

22、2020-10-13 LeetCode用队列实现栈，官方解答中用swap函数查一下。

Stack<int> st。st.push(x)函数为什么只需要一个参数就可以。如何计算时间复杂度，空间复杂度，要求时间空间复杂度怎么写代码。C++的reference网址

如果你想刷 LeetCode 的话，推荐一款 VS Code LeetCode 插件，可以方便你在编辑器里刷题。

<https://marketplace.visualstudio.com/items?itemName=LeetCode.vscode-leetcode>

LeetCode的题目可以在B站上看。

23、2020-10-15

vector<int> reversePrint(ListNode\* head) {

stack<int> st;

vector<int> res;

while(head){// push

st.push(head->val);

head = head->next;

}

while(!st.empty()){ // pop

res.push\_back(st.top());

st.pop();

}

return res;

}

res是一个动态数组，不是数组的首地址，有点像 int function(int a=5; return a)