

开放题和代码设计题

用栈或队列实现网页

两个栈：一个前进栈，一个后退栈。

前进操作：当前网页加入后退栈，从前进栈里取出前一个网页。

后退操作：当前网页加入前进栈，从后退栈里取出后一个网页。

新网页：**清空前进栈**，将当前网页压入后退栈。

中缀表达式转前缀表达式：

倒序遍历中缀表达式，

执行**中缀表达式转后缀表达式**的操作

最后翻转即可

计算前缀表达式的值：用栈实现。

循环队列：

注意栈顶指针指的是栈顶元素还是栈顶元素的下一个位置

判断队空和队满的区别

队空： $Q.front = Q.rear$?

队满：两种不同方式

空位法：

$(Q.rear + 1) \bmod size == Q.front \bmod size$

标记位法：

push时设置 $Tag = 1$,Pop时设置 $Tag = 0$.根据 $Tag \& \& Q.front == Q.rear ? 1 : 0$ 判断队列满或者队空。

01. 若希望循环队列中的元素都能得到利用，则需设置一个标志域 tag，并以 tag 的值为 0 或 1 来区分队头指针 front 和队尾指针 rear 相同时的队列状态是“空”还是“满”。试编写与此结构相应的入队和出队算法。

链队列：

链队列的优点,普通队列的缺点

普通队列的缺点：大小有限

链队列的优点：**大小无限**

链队列中加入新元素的位置：头插法还是尾插法？

反转列表

递归次数，基本条件

上台阶问题

动态规划: $dp[i]=dp[i-1]+dp[i-2]$

递归

硬币找零问题的优化

0-1背包问题

树的遍历

使用栈实现树的前序、中序和后序遍历

栈起到了回溯的作用,并不像BFS中队列记录即将访问的元素.

408 大题(*代表难度)

共享栈*

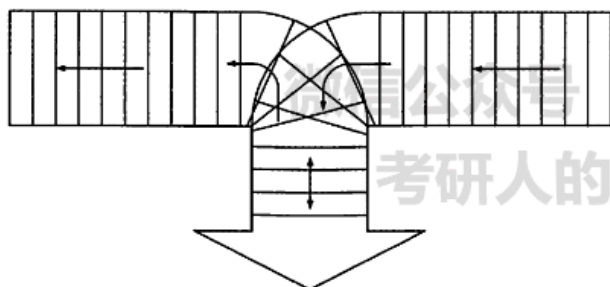
05. 设有两个栈 s_1 、 s_2 都采用顺序栈方式，并共享一个存储区 $[0, \dots, \text{maxsize}-1]$ ，为了尽量利用空间，减少溢出的可能，可采用栈顶相向、迎面增长的存储方式。试设计 s_1 、 s_2 有关入栈和出栈的操作算法。

只增加空间的栈**

- 链栈
 - 双指针:头指针和尾指针
 - 出栈:尾指针的下一个元素等于头指针,头指针指向元素值设置为无效值并清空指向,且指向下一个元素作为队列头.更新尾指针.
 - 入栈:判断尾指针指向的元素是否为空,如果为空,则更新加入的元素值.否则,新建结点,尾指针指向下一个元素,更新尾指针.
04. 【2019 统考真题】请设计一个队列，要求满足：①初始时队列为空；②入队时，允许增加队列占用空间；③出队后，出队元素所占用的空间可重复使用，即整个队列所占用的空间只增不减；④入队操作和出队操作的时间复杂度始终保持为 $O(1)$ 。请回答下列问题：
- 1) 该队列是应选择链式存储结构，还是应选择顺序存储结构？
 - 2) 画出队列的初始状态，并给出判断队空和队满的条件。
 - 3) 画出第一个元素入队后的队列状态。
 - 4) 给出入队操作和出队操作的基本过程。

火车调度问题*

02. 按下图所示铁道进行车厢调度（注意，两侧铁道均为单向行驶道，火车调度站有一个用于调度的“栈道”），火车调度站的入口处有 n 节硬座和软座车厢（分别用 H 和 S 表示）等待调度，试编写算法，输出对这 n 节车厢进行调度的操作（即入栈或出栈操作）序列，以使所有的软座车厢都被调整到硬座车厢之前。



分析与关键点

1. 问题特点：

- 输入是一个包含硬座车厢（H）和软座车厢（S）的序列。
- 初始车厢顺序不满足要求，需要调整成所有软座车厢（S）排在硬座车厢（H）前。
- 通过一个辅助的“栈道”（栈结构）来完成调整。

2. 栈的特点：

- 栈的操作是“后进先出”（LIFO）。
- 在调整过程中，栈可以暂存车厢，按需要重新输出，从而改变车厢顺序。

3. 解法思路：

- 遍历输入车厢序列，对每个车厢进行操作：
 1. 如果是软座车厢（S），直接输出。
 2. 如果是硬座车厢（H），入栈暂存。
- 遍历结束后，将栈中所有车厢依次出栈输出（保持原硬座车厢的顺序）。
- 通过这种方式，保证软座车厢先输出，而硬座车厢后输出。

栈实现递归**

自顶向上,栈充当临时变量的作用.

03. 利用一个栈实现以下递归函数的非递归计算:

微信公众号: 神灯考研

客服微信: KYFT104

QQ群: 118105451

关注微信公众号【神灯考研】，获取更多考研资料 第3章 栈、队列和数组 093

$$P_n(x) = \begin{cases} 1, & n=0 \\ 2x, & n=1 \\ 2xP_{n-1}(x) - 2(n-1)P_{n-2}(x), & n>1 \end{cases}$$

解题思路

我们需要使用栈来实现给定递归函数 $P_n(x)$ 的非递归计算。这是一个典型的递归转非递归问题，核心思想是利用栈来模拟递归调用过程。

递归公式分析

递归公式定义如下:

$$P_n(x) = \begin{cases} 1, & \text{当 } n=0 \\ 2x, & \text{当 } n=1 \\ 2x \cdot P_{n-1}(x) - 2(n-1) \cdot P_{n-2}(x), & \text{当 } n>1 \end{cases}$$

特点分析:

1. $P_n(x)$ 的值依赖于两个前置值 $P_{n-1}(x)$ 和 $P_{n-2}(x)$ 。
2. 递归公式可以转化为自底向上计算, 即从 $P_0(x)$ 和 $P_1(x)$ 开始, 逐步计算到 $P_n(x)$ 。
3. 为了实现非递归计算, 可以使用栈来模拟递归调用过程, 或者直接用循环配合临时变量完成。

04. 某汽车轮渡口, 过江渡船每次能载 10 辆车过江。过江车辆分为客车类和货车类, 上渡船有如下规定: 同类车先到先上船; 客车先于货车上船, 且每上 4 辆客车, 才允许放上 1 辆货车; 若等待客车不足 4 辆, 则以货车代替; 若无货车等待, 允许客车都上船。试设计一个算法模拟渡口管理。

WPL带权路径长度*

- DFS/BFS都行

19. 【2014 统考真题】二叉树的带权路径长度 (WPL) 是二叉树中所有叶结点的带权路径长度之和。给定一棵二叉树 T , 采用二叉链表存储, 结点结构为

left	weight	right
------	--------	-------

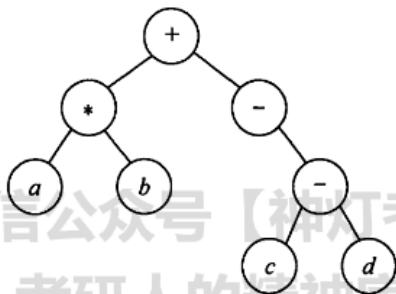
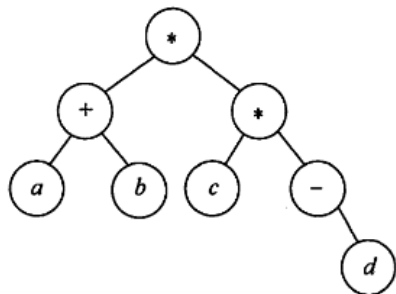
其中叶结点的 weight 域保存该结点的非负权值。设 root 为指向 T 的根结点的指针, 请设计求 T 的 WPL 的算法, 要求:

- 1) 给出算法的基本设计思想。
- 2) 使用 C 或 C++ 语言, 给出二叉树结点的数据类型定义。
- 3) 根据设计思想, 采用 C 或 C++ 语言描述算法, 关键之处给出注释。

根据解析树构造中序表达式*

- 后序遍历, 遇到运算数时加上 (即可

20. 【2017 统考真题】请设计一个算法, 将给定的表达式树 (二叉树) 转换为等价的中缀表达式 (通过括号反映操作符的计算次序) 并输出。例如, 当下列两棵表达式树作为算法的输入时:



输出的等价中缀表达式分别为 $(a+b) * (c * (-d))$ 和 $(a*b) + (- (c-d))$ 。

判断二叉搜索树是否有效*

- 中序遍历

```
left=dfs(root->left); //bool 子树是否有序
right=dfs(root->right);
return root->val<root->right->val
&&root->val>root->left->val
&&left&&right// 当前树是否有序,子树是否有序?
```

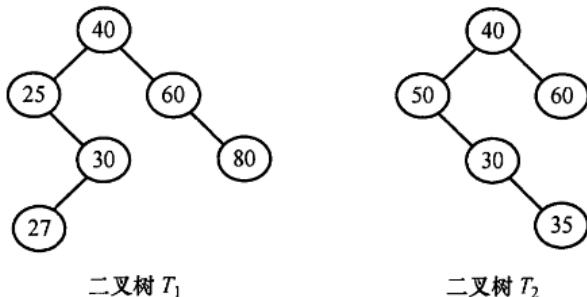
- 比较根节点与左子树根节点的大小,左子树根节点小于根节点返回True,否则返回False

- 其实就是一个树形dp

21. 【2022 统考真题】已知非空二叉树 T 的结点值均为正整数，采用顺序存储方式保存，数据结构定义如下：

```
typedef struct {           // MAX_SIZE 为已定义常量
    int SqBiTNode[MAX_SIZE]; // 保存二叉树结点值的数组
    int ElemNum;             // 实际占用的数组元素个数
} SqBiTree;
```

T 中不存在的结点在数组 SqBiTNode 中用 -1 表示。例如，对于下图所示的两棵非空二叉树 T_1 和 T_2 ,



T_1 的存储结果如下：

T1.SqBiTNode	40	25	60	-1	30	-1	80	-1	-1	27		
T1.ElemNum = 10												

T_2 的存储结果如下：

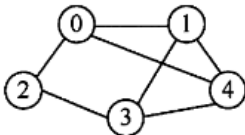
T2.SqBiTNode	40	50	60	-1	30	-1	-1	-1	-1	-1	35	
T2.ElemNum = 11												

请设计一个尽可能高效的算法，判定一棵采用这种方式存储的二叉树是否为二叉搜索树，若是，则返回 true，否则，返回 false。要求：

- 1) 给出算法的基本设计思想。
- 2) 根据设计思想，采用 C 或 C++ 语言描述算法，关键之处给出注释。

图的属性*

05. 【2015 统考真题】已知含有 5 个顶点的图 G 如下图所示。



请回答下列问题：

- 1) 写出图 G 的邻接矩阵 A (行、列下标从 0 开始)。
- 2) 求 A^2 ，矩阵 A^2 中位于 0 行 3 列元素值的含义是什么？
- 3) 若已知具有 n ($n \geq 2$) 个顶点的图的邻接矩阵为 B ，则 B^m ($2 \leq m \leq n$) 中非零元素的含义是什么？

EL边*

抽象题目

- 统计度数并检查是否有奇数边

06. 【2021 统考真题】已知无向连通图 G 由顶点集 V 和边集 E 组成, $|E| > 0$, 当 G 中度为奇数的顶点个数为不大于 2 的偶数时, G 存在包含所有边且长度为 $|E|$ 的路径 (称为 EL 路径)。设图 G 采用邻接矩阵存储, 类型定义如下:

```
typedef struct{                                //图的定义
    int numVertices,numEdges;                  //图中实际的顶点数和边数
    char VerticesList[MAXV];                  //顶点表。MAXV 为已定义常量
    int Edge[MAXV][MAXV];                     //邻接矩阵
}MGraph;
```

请设计算法 `int IsExistEL(MGraph G)`, 判断 G 是否存在 EL 路径, 若存在, 则返回 1, 否则返回 0。要求:

- 1) 给出算法的基本设计思想。
- 2) 根据设计思想, 采用 C 或 C++ 语言描述算法, 关键之处给出注释。
- 3) 说明你所设计算法的时间复杂度和空间复杂度。

是否为Tree*

+ Tree:连通无环图

并查集

02. 试设计一个算法, 判断一个无向图 G 是否为一棵树。若是一棵树, 则算法返回 true, 否则返回 false。

图的遍历

BFS可以解决存在性问题,路径种类和路径名称不太好解决.

DFS更加方便.

04. 分别采用基于深度优先遍历和广度优先遍历算法判别以邻接表方式存储的有向图中是否存在由顶点 v_i 到顶点 v_j 的路径 ($i \neq j$)。注意, 算法中涉及的图的基本操作必须在此存储结构上实现。
05. 假设图用邻接表表示, 设计一个算法, 输出从顶点 V_i 到顶点 V_j 的所有简单路径。