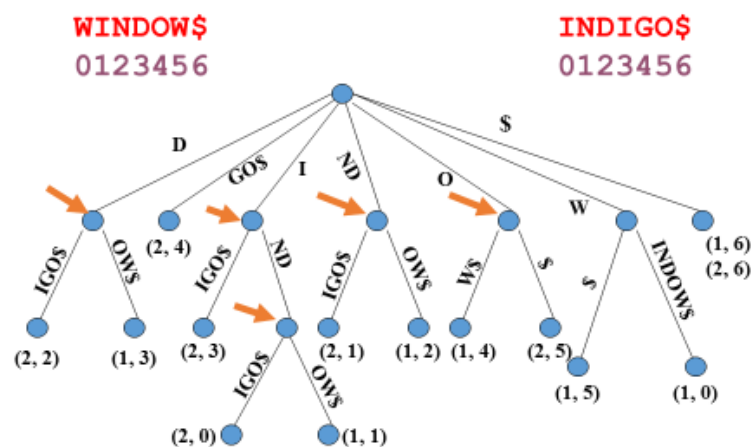


1. $A[N][N]$ 是对称矩阵, 现将下三角矩阵按行存储到一维数组 $T[N(N+1)/2]$ 中(包括对角线), 则对任一上三角元素 $A[i][j]$ 其对应值 $(0 \leq i \leq j < N)$ 在 $T[k]$ 中的下标 k 是_____
2. (共 5 分) 已知广义表: $A=(a, b)$, $B=(A, A)$, $C=(c, (d, A), B, e)$
 - (1) (2 分) 画出广义表 C 带表头的存储结构图
 - (2) (2 分) 广义表 C 的深度和长度分别为
 - (3) (1 分) $\text{tail}(\text{head}(\text{tail}(C)))$ 的运算结果为
3. 下面是字符串 WINDOW 和 INDIGO 的所有前缀串组成的后缀树示意图(压缩了单路径的 Trie)。请回答以下问题
 - (1) 请描述在这棵后缀树中, 寻找两个字符串的最长公共前缀子串 (Longest Common Prefix, 简称 lcp) 的算法思想。lcp(i, j), i 和 j 分别为第一个串和第二个串起始下标, Lcp 是指从下标 i (或 j) 开始到字符串结尾对应字符串的最大公共前缀。例如, 对于 WINDOW 和 INDIGO 两个串, $\text{lcp}(0, 1) = ""$; $\text{lcp}(1, 0) = "IND"$ 。
 - (2) 请阐述加特殊符号 '\$' 的理由



4. 在一棵空 AVL 树中, 顺序插入如下关键词: {5, 9, 4, 2, 1, 3, 8}。
 - (1) 严格遵循 AVL 树的操作规定, 画出每次插入后得到的 AVL 树。
 - (2) 全部插入后, 求等概率下查找成功的平均检索长度。
5. 现有一个工资系统, 请实现满足如下操作需求的 Splay 树, 并给出算法的伪代码: (可以自定义一些全局变量)
 - (1) `void insert(int w);` 新加一位工资为 w 的员工 (假设没有重复值)
 - (2) `void add(int d);` 每位员工都加上工资 d
 - (3) `void fire(int t);` 解雇工资大于 t 的员工

伸展树是一种自平衡的 BST, 数据结构如下:

```
struct TreeNode {
    int data;
    TreeNode * father, * left, * right;
};
```

可以直接使用的函数:

```
void Splay(TreeNode* x, TreeNode* f); // 将 x 旋转为 f 的子结点 (f 在 x 的祖先路)
// 把 x 旋转到根结点即 Splay(x, NULL)
```

```
TreeNode * find(int x, TreeNode* S); //表示在 S 中查找元素 x 的位置，若找不到，返回其该插入位置的父亲结点  
void Delete(TreeNode* x); // 删除以 x 结点为根的子树
```