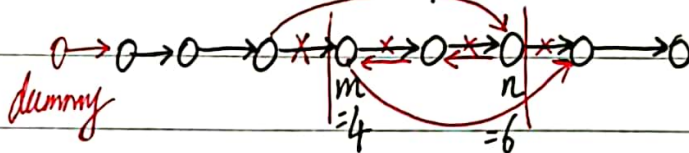


91. 解码方法: dp问题. 类似台阶问题.

① 状态表示: $f(i)$ 表示由前 i 个字符解码可得到字符串的个数.

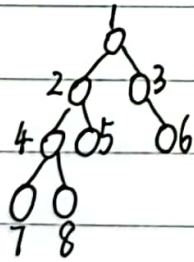
② 状态计算: $f(i) = \begin{cases} f(i-1) + f(i-2), & \text{if } s[i-1] \neq 0 \text{ and } s[i-1]s[i-2] \leq 26 \\ f(i-1), & \text{if } s[i-1] \neq 0 \text{ and } s[i-1]s[i-2] > 26 \\ f(i-2), & \text{if } s[i-1] = 0 \text{ and } s[i-2] \neq 0 \end{cases}$ 细节见代码!

92. 反转链表 II. 思路: 

93. 复原IP地址. 简单的dfs搜索问题.
 { ① 0~255
 ② 无前导0
 ③ 无非数字字符. } 详见代码.

94. 二叉树的中序遍历: 递归法 (很简单略)

迭代法: 用栈模拟, 举例说明:



stack: 1 2 4 7

→ 1 2 4

→ 1 2

→ 1 2 8

→ 1 2

→ 1

→ 1 5

→ 1

→

→ 3

→

→ 6

→

遍历结果

7

7 4

7 4 8

7 4 8 2

7 4 8 2 5

7 4 8 2 5 1

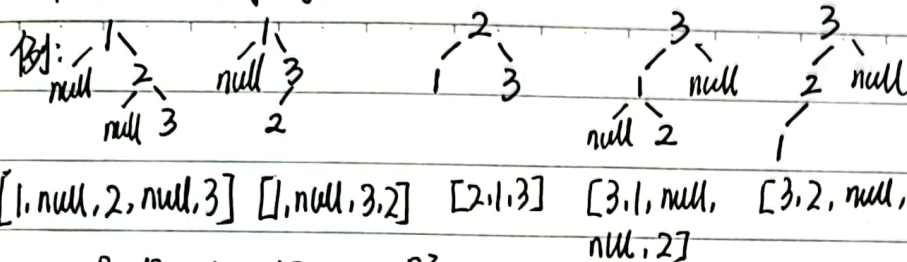
7 4 8 2 5 1 3

7 4 8 2 5 1 6

★ 详细过程见代码.

95. 不同的二叉搜索树II: 方案数 $Catalan(n) = \frac{C_{2n}^n}{n+1} = \frac{(2n)!}{n!(n+1)!}$.

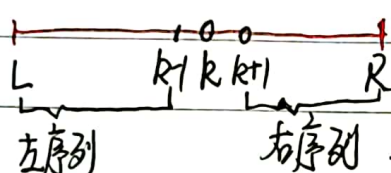
Date: / /



递归代码.

利用爆搜的方法解决此题.

★ 枚举根节点, 其左子树与右子树的序列都是连续的. (利用该性质进行递归)



利用乘法原理: 左边 x 个方案, 右边序列 y 个方案. 则共有 $x \cdot y$ 个方案.

96. 不同的二叉搜索树: 方法一: 用95题的递归方法 (记忆化搜索) 方法二: 利用 $Catalan(n)$ 的通项公式. 详见代码

97. 交错字符串: 含义: 在 S_3 中找到两个子序列, 其中一个子序列是 S_1 , 另一个是 S_2 . 方法: 使用 dp 的方法做.

① 状态表示: $f(i, j)$ 表示能否由 $S_1[1 \sim i]$ 和 $S_2[1 \sim j]$ 交错形成 $S_3[1 \sim i+j]$

② 状态计算: $f(i, j) = \begin{cases} f(i-1, j), & S_3[i+j] == S_1[i] \\ f(i, j-1), & S_3[i+j] == S_2[j] \\ \text{false}, & \text{其他} \end{cases}$

98. 验证二叉搜索树: 递归每个节点的最大值和最小值, 详见代码.

99. 恢复二叉搜索树: ① 观察: 有两种情况 (i) 相邻中序遍历相邻两数交换.

如: 1 2 3 4 5 \rightarrow 1 3 2 4 5

(ii) 中序遍历不相邻两数交换: 如: 1 2 3 4 5 6 7 \rightarrow 1 6 3 4 5 2 7

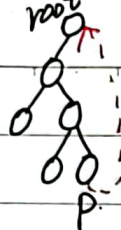
② 算法: 找中序遍历中相邻数字形成逆序对. 若这样的逆序对只有1个, 则只需交换这两个数字. 若有两个, 交换第一个逆序对的第一个和第二个逆序对的第二个数字.

③ 为了满足只使用 $O(1)$ 空间的要求, 需使用 Morris 遍历算法. (其实是对使用栈进行中序遍历的改进)

时间复杂度为 $O(n)$ (每个节点遍历2次)

Morris 遍历介绍: ① 没有左子树, 则直接遍历该点, 然后走到右儿子

② 有左子树,



则找前驱节点 P,

(中序遍历的)

② 否则,

$p \rightarrow right = NULL$,

遍历 root,

$root = root \rightarrow right$.

① 若 $p \rightarrow right = NULL$, 则令

$p \rightarrow right = root$.

$root = root \rightarrow left$.

$root = root \rightarrow right$.

我的方法: 先从 root 走到 $root \rightarrow left$, 然后从该点一直向右走即可.

模拟的以上过程的例子见 [github](#).

100. 相同的树: 同时遍历两棵树即可 详见代码

101. 对称二叉树: 方法一: 递归: 对称地遍历子树, 详见代码.

方法二: 迭代: 思路与详情见代码. (另参考 94 题)

102. 二叉树的层次遍历: 借助队列搜索即可. bfs 搜索的思路, 详见代码

103. 二叉树的锯齿形层序遍历: 即 dfs

复习: 树的遍历 / 前中后序遍历: 3 种方法 (递归, 用栈模拟递归, morris 遍历)

层次遍历: 用队列 bfs 即可.

回到本题: 代码框架与 102 相同, 只需加一步: 偶数层的数字应翻转一下.

104. 二叉树的最大深度: 方法一: 使用 bfs 的层次遍历 (与 102. 103 思路相同)

方法二: 使用 dfs.

root 处的深度为: l, r 中深度较大者的深度加 1.

105. 从前序与中序遍历序列构造二叉树.

步骤: ① 由前序遍历找根节点, ② 在中序遍历中找到该根节点, 由此可知左子树, 右子树的大小.

③ 递归如上步骤

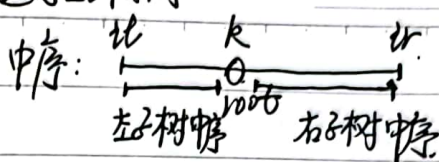
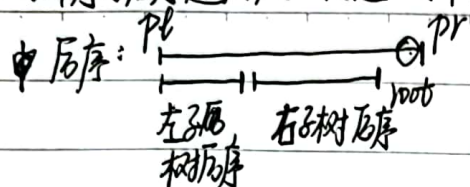
(利用哈希表加速查找)

前序: pre 根 左子树前 右子树前.

中序: in 左子树中 根 右子树中.

★ 详见代码.

106. 从中序与后序遍历序列构造二叉树: 与上一题思路相同.



107. 二叉树的层序遍历 II: 与102的思路相同, 只需将答案 reverse-T.

108. 将有序数组转换为二叉搜索树: 实际上考查了线段树和平衡树的初始化方式.



取中点为根, 左右两侧分别为左、右子树, 如此递归进行即可.

证明上述方法的正确性: 设区间长度为 n , 可证明, 用如上方法构造出的二叉搜索树的高度为: $\lceil \log_2(n+1) \rceil$, 现证明该结论.

分两部分来证明: ① 一定有: $h \geq \lceil \log_2(n+1) \rceil$, 这是因为 n 个结点, 高度最低的树是完全二叉树, 而其高度 ~~等于~~ $\lceil \log_2(n+1) \rceil$.

② 现证明 $h \leq \lceil \log_2(n+1) \rceil$. 利用数学归纳法证明.

($n > 0$) (A) 若 n 是奇数, 则左右子树都有 $\frac{n-1}{2}$ 个节点, 根据归纳假设左右子树高度均为: $\lceil \log_2(\frac{n-1}{2} + 1) \rceil$.

则构造出二叉树的高度为: $\lceil \log_2(\frac{n-1}{2} + 1) \rceil + 1$, 现只需证明:

$$\lceil \log_2(\frac{n-1}{2} + 1) \rceil + 1 \leq \lceil \log_2(n+1) \rceil.$$

证: $\lceil \log_2(\frac{n-1}{2} + 1) \rceil + 1 = \lceil \log_2(\frac{n-1}{2} + 1) \rceil + 1 \therefore$ 只需证:

$$\lceil \log_2(\frac{n-1}{2} + 1) \rceil + 1 \leq \lceil \log_2(n+1) \rceil. \quad \text{而: } \log_2(\frac{n-1}{2} + 1) + 1 = \log_2(n+1)$$

\therefore 得证.

($n > 0$) (B) 若 n 是偶数. 不妨设左子树节点数为 $\frac{n-2}{2}$, 右子树为 $\frac{n}{2}$

由归纳假设, 左子树高度为 $\log_2(\frac{n-2}{2} + 1)$, 右子树为 $\log_2(\frac{n}{2} + 1)$

\therefore 只需证明: $\lceil \log_2(\frac{n}{2} + 1) \rceil + 1 \leq \lceil \log_2(n+1) \rceil$

即: $\lceil \log_2(n+2) \rceil \leq \lceil \log_2(n+1) \rceil$, 事实上, 有 $\lceil \log_2(n+2) \rceil = \lceil \log_2(n+1) \rceil$

反证: 设一正整数 k , 若成立, 则有: $\log_2(n+2) > k$ 且 $\log_2(n+1) \leq k$.

于是: $n+2 > 2^k \geq n+1$ $\because n$ 是偶数 $\therefore n+2 > 2^k > n+1$, 而 $n+2$ 与 $n+1$ 之间不能存在一个新的整数 2^k , 矛盾! $\therefore \lceil \log_2(n+2) \rceil = \lceil \log_2(n+1) \rceil$.

109. 有序链表转换二叉搜索树.

Date. /

方法一: 利用额外的 $O(n)$ 空间开一数组, 转化为上一问题

X 方法二: 快慢指针, 该方法略, 比较慢, 时间 $O(n \log n)$ (没看该方法)

方法三: 利用中序遍历优化, 详见代码.

赋值相当于中序遍历中的打印值.

②创建
③赋值



①创建
④赋值

⑤创建

⑥赋值.

110. 平衡二叉树.

思路: 递归进行, 递归函数返回根节点, 左 & 右子树的高度.

详见代码