Week2

1.树

二叉树：每个节点只有两个分支，没有环。

图：有环。

Linked List Linked List 是特殊化的 Tree，Tree 是特殊化的 Graph。

1.1树的节点定义

代码要记住，变成机械化记忆。

Python

```
class TreeNode:
  def __init__(self,val):
    self.val = val
    self.left, self.right = None,None
```

Java

```
public class TreeNode {
  public int val;
  public TreeNode left, right;
  public TreeNode(int val) {
    this.val = val;
    this.left = null;
    this.right = null;
  }
}
```

1.2二叉树遍历：基于递归

1. 前序（Pre –order）：根 –左–右
2. 中序（In –order）：左 –根–右
3. 后序（Post Post –order）：左 –右–根

代码要记住，变成机械化记忆。

```
def preorder(self,root):
  if root:
    self.traverse_path.append(root.val)
    self.preorder(root.left)
    self.preorder(root.right)

```

```
def inorder(self,root):
  if root:
    self.inorder(root.left)
    self.traverse_path.append(root.val)
    self.inorder(root.right)

def postorder(self,root):
  if root:
    self.postorder(root.left)
    self.postorder(root.right)
    self.traverse_path.append(root.val)
```

1.3二叉搜索树

二叉搜索树是指一棵空树或者具有下列性质的二叉树：

1. 左子树上 所有结点 的值 均小于它根结点；

2. 右子树上 所有结点 的值 均大于它根结点；

3. 以此类推：左、右子树 也分别为二叉查找。 （这就是 重复性 ！）

中序遍历：升排列

查询、删除的时间复杂度都是O(logn)

创建二叉搜索树，就是先建一颗空树，然后不断插入节点。

插入：先查找，没查到就插入

删除：如果是叶子节点，直接删除；

如果根节点(假设删65)，找离65最近的节点来顶替，一般取第一个大于65的节点，即右子树最小的节点。

Demo：

https://visualgo.net/zh/bst