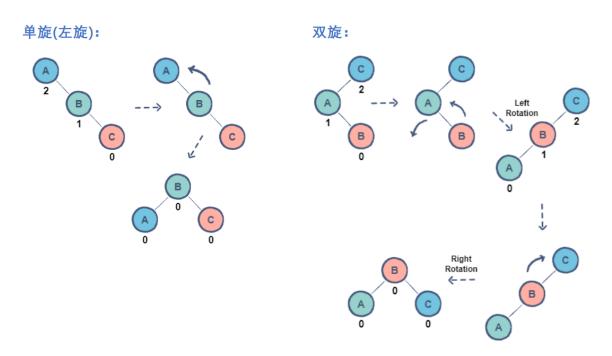
Homework 1: Warm-up

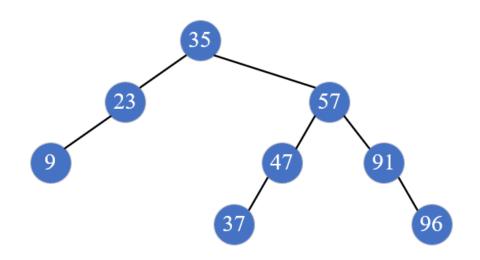
Part 1: AVL Tree 简介

AVL 树是一种自平衡二叉搜索树,其构造的规则是:任何节点的两个子树的**高度最大差别为 1**。为了维持 AVL 树的高度差,在插入节点时,AVL 树可能会进行旋转操作。根据旋转的方向我们可以分为两大种类型: **单旋**(向左旋、向右旋)以及**双旋**(先左旋后右旋、先右旋后左旋),如下图所示(节点下方的数字表示节点左右子树的高度之差)。



Exercise 1:

画出下方 AVL Tree 依次插入序列 5, 42, 51 后的结果:



Part 2: 实践部分

助教在 Canvas 已给出 AVL Tree 实现的部分代码(缺省的函数及BST的实现需自行补充),可供参考。

Exercise 2:

补全 AVL Tree 和 BST 的相关代码(*你也可以自行实现*),并**设计实验**简单证明 AVL Tree 和 BST 的性能差异。

(例如:随机生成一组插入序列,插入元素的个数分别为 50, 100, 500, 1000, 2000, 比较插入后 AVL Tree 和 BST 的树高或查找操作的时间)

Part 3: 注意事项

- 请将你对于各个 Exercise 问题的回答和实验部分的结果和分析写成报告提交,与源代码和输入样例 (如有) 一起做成压缩包上传到 Canvas。命名使用"学号+姓名+hw1",如"521123456789+张三 +hw1.zip"。
- **请勿抄袭**!课后作业采用倒扣分制,如果有遗漏或者得分不足会在最终成绩酌情减分,同时课后作业的内容会体现在期末试卷中,对同学们也是一种练习。
- 本次作业的截止时间是 **2023年2月26日 23:59**, 迟交将会酌情扣分。
- 有任何作业相关的问题可以询问 熊天磊、江玙璠 助教。