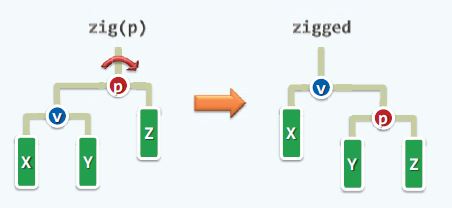
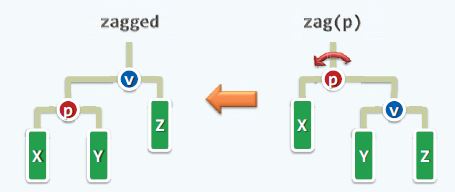
**可以这样来记忆：zig是顺时针(clockwise)方向，zag是逆时针(anti-clockwise)方向；而zig(p)或zag(p)可以形象地看成是“把p压下来，把它的孩子翘上去”。**





这样的zig/zag局部拓扑结构调整，在实现步骤上可以分如下三步走，代码上面有。

　　（1）v与p的祖先g建立连接 --->（2）v的孩子Y过继给p   ---> （3）v与p角色互换

伸展树(Splay Tree)是一种二叉排序树，它能在O(log n)内完成插入、查找和删除操作。它由Daniel Sleator和Robert Tarjan创造。

(01) 伸展树属于二叉查找树，即它具有和二叉查找树一样的性质：假设x为树中的任意一个结点，x节点包含关键字key，节点x的key值记为key[x]。如果y是x的左子树中的一个结点，则key[y] <= key[x]；如果y是x的右子树的一个结点，则key[y] >= key[x]。

(02) 除了拥有二叉查找树的性质之外，伸展树还具有的一个特点是：当某个节点被访问时，伸展树会通过旋转使该节点成为树根。这样做的好处是，下次要访问该节点时，能够迅速的访问到该节点。

假设想要对一个二叉查找树执行一系列的查找操作。为了使整个查找时间更小，被查频率高的那些条目就应当经常处于靠近树根的位置。于是想到设计一个简单方法，在每次查找之后对树进行重构，把被查找的条目搬移到离树根近一些的地方。伸展树应运而生，它是一种自调整形式的二叉查找树，它会沿着从某个节点到树根之间的路径，通过一系列的旋转把这个节点搬移到树根去。

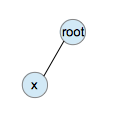
相比于"二叉查找树"和"AVL树"，学习伸展树时需要重点关注是"伸展树的旋转算法"。

树的搜索效率与树的深度有关。二叉搜索树的深度可能为n，这种情况下，每次搜索的复杂度为n的量级。AVL树通过动态平衡树的深度，单次搜索的复杂度为log(n) (以上参考[纸上谈兵 AVL树](http://www.cnblogs.com/vamei/archive/2013/03/21/2964092.html))。我们下面看伸展树(splay tree)，它对于m次连续搜索操作有很好的效率。

 伸展树会在一次搜索后，对树进行一些特殊的操作。这些操作的理念与AVL树有些类似，即通过旋转，来改变树节点的分布，并减小树的深度。但伸展树并没有AVL的平衡要求，任意节点的左右子树可以相差任意深度。与二叉搜索树类似，伸展树的单次搜索也可能需要n次操作。**但伸展树可以保证，m次的连续搜索操作的复杂度为mlog(n)的量级，而不是mn量级。**

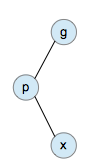
**具体来说，在查询到目标节点后，伸展树会不断进行下面三种操作中的一个，直到目标节点成为根节点 （注意，祖父节点是指父节点的父节点）**

1. zig: 当目标节点是根节点的左子节点或右子节点时，进行一次单旋转，将目标节点调整到根节点的位置。



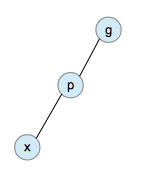
zig

2. zig-zag: 当目标节点、父节点和祖父节点成"zig-zag"构型时，进行一次双旋转，将目标节点调整到祖父节点的位置。



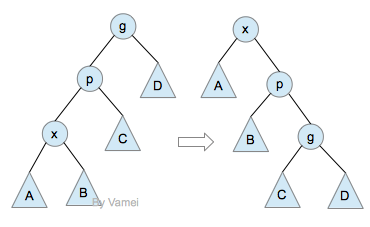
zig-zag

3. zig-zig：当目标节点、父节点和祖父节点成"zig-zig"构型时，进行一次zig-zig操作，将目标节点调整到祖父节点的位置。



zig-zig

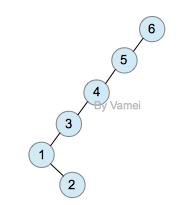
单旋转操作和双旋转操作见[AVL树](http://www.cnblogs.com/vamei/archive/2013/03/21/2964092.html)。下面是zig-zig操作的示意图:



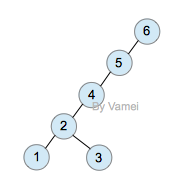
zig-zig operation

在伸展树中，zig-zig操作(基本上)取代了AVL树中的单旋转。通常来说，如果上面的树是失衡的，那么A、B子树很可能深度比较大。相对于单旋转(想一下单旋转的效果)，zig-zig可以将A、B子树放在比较高的位置，从而减小树总的深度。

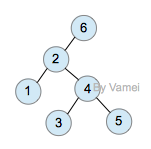
下面我们用一个具体的例子示范。我们将从树中搜索节点2：



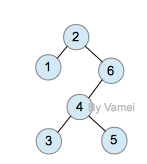
Original



zig-zag (double rotation)



zig-zig



zig (single rotation at root)

上面的第一次查询需要n次操作。然而经过一次查询后，2节点成为了根节点，树的深度大减小。整体上看，树的大部分节点深度都减小。此后对各个节点的查询将更有效率。

伸展树的另一个好处是将最近搜索的节点放在最容易搜索的根节点的位置。在许多应用环境中，比如网络应用中，某些固定内容会被大量重复访问(比如江南style的MV)。伸展树可以让这种重复搜索以很高的效率完成。