在这个程序中，将实现AVL树和一组与AVL树相关的函数。为了简单起见，我们做以下假设:

1. AVL树的每一项都包含一个整型键和一个整型值。

2. 没有AVL树包含重复的项目。当k1=k2和v1=v2保持不变时，两个项目(k1, v1)和(k2, v2)是重复的

3.一个AVLtree可以包含多个具有相同键的项。

任务：

1. AVLTree \*CreateAVLTree(const char \*filename)。

这个函数根据参数filename从文本文件或标准输入(键盘)中读取所有项，从而创建一个AVL树。如果filename是"stdin"，此函数将从标准输入中读取所有项。否则，它将从以filename作为完整路径名的文本文件中读取所有项。

注：一个输入文本文件包含零个或多个项，其中每个项的形式(键，值)。任何字符，如两个相邻项目之间的空白将被忽略。

例如，下面的示例文件包含3个项目。（2，50）（4，30）（9，30）

类似地，从标准输入读取时，每个输入行可以有零个或多个项，由一个或多个空格字符分隔。空行表示输入结束。如果输入中出现错误，这个函数将打印错误并终止程序。

您可以假设输入不包含重复的项，因此这个函数不需要检查重复的项。这个函数的时间复杂度不能高于O(n logn)，其中n是生成的AVL树的大小。如果您的时间复杂度较高，您将得到O标记的这个函数。可以假设对C内置函数的每次调用都需要O(1)时间。

1. AVLTree \* CloneAVLTree (AVLTree \* T)

这个函数对输入的AVL树T创建一个相同的复制(克隆)，并返回一个指向克隆树的指针。这个函数的时间复杂度不能高于O(n)，其中n是t的大小。

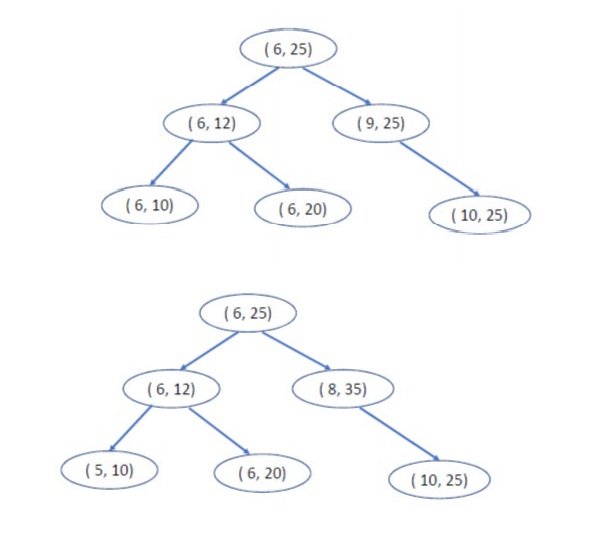
AVLTree \*AVLTreesUnion(AVLTree \*T1, AVLTree \*T2)

这个函数计算两个AVL树T1和T2的并集树，并返回一个指向并集树的指针。两个AVL树T1和T2的并集树是一个AVL树，它包含T1和T2的所有项，并且没有重复的项。假设T1和T2都不包含重复的项。

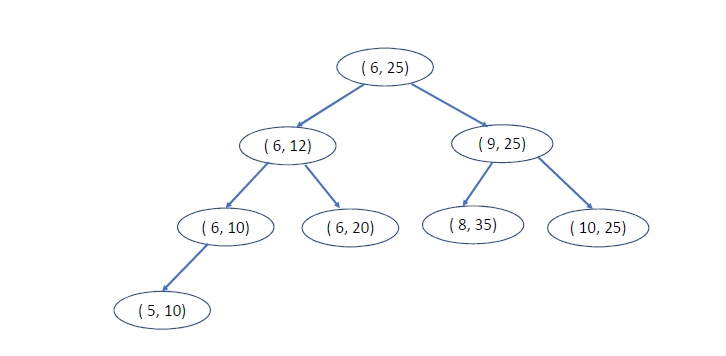
该函数的时间复杂度不能大于O(m+n)，其中m和n分别为T1和T2的大小。

例：

T1和T2分别为



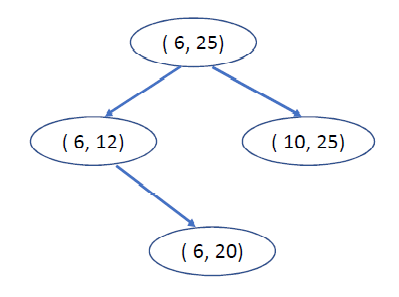
联合树（构造不唯一），取决于如何构造的



1. AVLTree \*AVLTreesIntersection(AVLTree \*T1, AVLTree \*T2)

这个函数计算两个AVL树T1和T2的交集树，并返回一个指向交集树的指针。两个AVL树T1和T2的交集树是一个包含所有在T1和T2中出现的项的AVL树。假设T1和T2都不包含重复的项。该函数的时间复杂度不能大于o(m+n)，其中m和n分别为T1和T2的大小，k为交集树的大小。

对于上述例子的交集树（构造可能不唯一）：



1. int InsertNode(AVLTree \*T, int k, int v)

如果项目(k, v)在树中存在，这个函数只返回0，而不将新的项目(k, v)添加到树中。否则，它将新项目(k, v)插入AVL树T中，将树的大小增加1并返回1。这个函数的时间复杂度不能高于O(log n)，其中n是t的大小。

int DeleteNode(AVLTree \*T, int k, int v).

如果该项(k, v)存在于AVL树T中，该函数删除包含该项的节点，将树的大小减少1并返回1。否则，它只返回0。这个函数的时间复杂度不能高于O(n)，其中n是t的大小。

1. AVLTreeNode \*Search(AVLTree \*T, int k, int v).

这个函数在AVL树T中搜索项目(k, v)。如果找到了项目，它返回一个指向包含该项目的节点的指针。否则，返回NULL。这个函数的时间复杂度不能高于O(n)，其中n是t的大小。

void FreeAVLTree(AVLTree \*T)

这个函数释放AVL树t所占用的堆空间。这个函数的时间复杂度不能高于O(n)，其中n是t的大小。可以假设每次调用free()都需要O(1)时间。

void PrintAVLTree (AVLTree \* T)

这个函数在标准输出(屏幕)上打印存储在AVL树T中的所有项，按键的非降序排序。每个项由(键，值)表示，每行有一个项。这个函数的时间复杂度不能高于O(n)，其中n是t的大小。您可以假设每次对内置C函数的调用都需要O(1)时间。

注：对于每个函数，分析其时间复杂度，并将时间复杂度分析作为注释放在函数前面。对于每个函数的时间复杂度，只需要给出主要组件(循环)的时间复杂度和总时间复杂度，且可以假设每次对内置C函数的调用都需要常数(O(1))时间。