**private void** splitNonLeafNode(Node ancestorUpper,Node ancestorLower,IntArrayList dpList){  
 **/…/**计算分裂后节点节点大小  
 IntArrayList rightList = **new** IntArrayList(rightSize);  
 **for**(**int** i = leftSize;i < dpCount;i++){  
 rightList.add(dpList.get(i));  
 }  
 dpList.removeRange(leftSize, dpCount - 1);  
 //更改左右节点的数据点集合

Node leftNode = **new** Node(nextNodeId(),**false**);  
 leftNode.setParent(ancestorUpper);  
 leftNode.setNodeHeight(ancestorUpper.getNodeHeight() + 1);  
 makeVpTree(leftNode,dpList);

//设置左节点，并以左节点为根节点，分裂新的子树  
  
 Node rightNode = **new** Node(nextNodeId(),**false**);  
 rightNode.setParent(ancestorUpper);  
 rightNode.setNodeHeight(ancestorUpper.getNodeHeight() + 1);  
 makeVpTree(rightNode,dpList);  
 //设置右节点，并以右节点为根节点，分裂新的子树

**int** pos = locateChildPos(ancestorUpper,ancestorLower);  
 //定位下方祖先节点的分支位置shiftBranchInfo(ancestorUpper,pos + 1,1);  
 *//*将后续分支的配置信息向后挪动/…../更新上方祖先节点的元数据}