```
private void splitLeafNode(Node correctLeafNode,Node parent){
/………./获取子节点列表代码
float[] newDistances = Arrays.copyOf(distances,childCount+1);
distances=newDistances;
insertIntoOrderedFloatArray(distances,newDistance,childCount);
//把子节点数组和距离数组插入对齐
childIds.insert(pos,id);

int allSize = childIds.size();
int leftSize = (int) Math.ceil(allSize * 1.0 / 2);//原则:左大右小
int rightSize = allSize-leftSize;
Node rightLeaf=new Node(nextNodeId(),true);
rightLeaf.initAsLeaf(rightSize,configuration.getEntrySize());
//初始化新的右侧的叶节点

childIds.removeRange(leftSize, childCount);
//清空分裂出去的位置

float branchDistances = parent.distances;
FloatArrayList branchBounds = parent.childrenBounds;

for(int i = branchCount - 1;i > branchPos;i--){
   childrenNodeIds[i + 1] = childrenNodeIds[i];
   branchDistances[i + 1] = branchDistances[i];
}
//移动 parent 的所有后续分支的槽位

childrenNodeIds[branchPos + 1] = rightLeaf.getId();
branchDistances[branchPos + 1] = rightLeaf.distances[rightSize
- 1];
branchBounds.insert(branchPos * 2, rightLeaf.distances[0]);
branchBounds.insert(branchPos * 2 + 1,
rightLeaf.distances[rightSize - 1]);
//更新新节点槽位的数据,这里的分支上下界可以直接插入，ArrayList 自动后移

nodePool.addNode(rightLeaf);
//将 rightLeaf 加入节点池中
```