**private void** redistributeLeafNode(correctLeafNode, parent) **{**

**int** bestBranch = -1;   
**int** largestDis = -1;//①记录最大未满值和最佳目标重分布分支

**int** left = branchPos - 1;  
**int** right = branchPos + 1; //②搜索起始位置，中心点两侧

**int** leftDPCount = 0;  
**int** rightDPCount = 0;*//*③记录左右重分布区间的数据点数量

**while**( right < branchCount || left >= 0){  
 **/……./** ④跟踪更新leftDPCount和rightDPCount  
 **if**(bestBranch != -1) **break**;*//*尽快结束搜索  
 left--; right++;//左右指针向两侧移动  
}

**if**(bestBranch !=-1){  
 **int** redisBranchCount = branchPos - bestBranch + 1; // ⑤

**int** dpCount = (leftDPCount \* 1.0) / redisBranchCount);  
 /……../ ⑥最好分支在左侧，各分支数据点依次向左移动，元数据的更新 /……../最好分支在右侧，分支数据点依次向右移动，元数据的更新

/……../⑦更新parent节点的界标数组  
 }

}

}