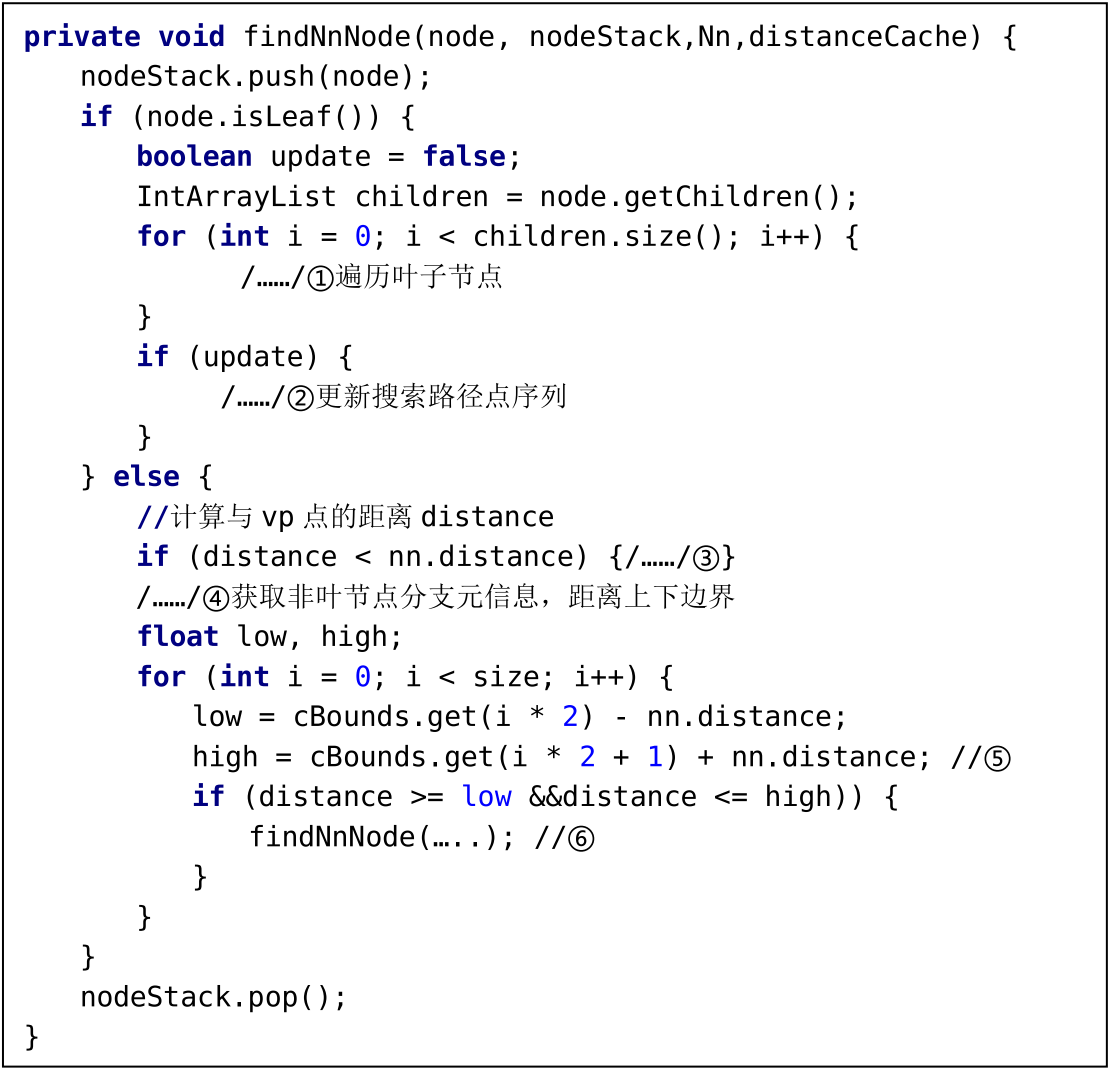

```
private void findNnNode(node, nodeStack,Nn,distanceCache) {
    nodeStack.push(node);
    if (node.isLeaf()) {
        boolean update = false;
        IntArrayList children = node.getChildren();
        for (int i = 0; i < children.size(); i++) {
               /……/①遍历叶子节点
        }
        if (update) {
              /……/②更新搜索路径点序列
        }
    } else {
        //计算与 vp 点的距离 distance
        if (distance < nn.distance) {/……/③}
        /……/④获取非叶节点分支元信息，距离上下边界
        float low, high;
        for (int i = 0; i < size; i++) {
            low = cBounds.get(i * 2) - nn.distance;
            high = cBounds.get(i * 2 + 1) + nn.distance; //⑤
            if (distance >= low &&distance <= high)) {
                findNnNode(…..); //⑥
            }
        }
    }
    nodeStack.pop();
}
```