NJ 算法（原始定义版）：

1. 将所有类初始化为一棵星状树，即每个分类单元成一类。
2. 对所有的分类单元，计算其到主中心的距离（枝长）为：

$$d_u = \frac{1}{n-2}\sum_{i \neq u} d_{ui}$$

3. 如果有三个以上的分类存在，则继续执行循环：
   a) 选择一对分类单元 x 与 y，使 $d_{xy} - d_x - d_y$ 最小
   b) 将 x 与 y 归并为新类 z，在星形树中添加一个新节点 z，
   c) 新节点 z 代表新生成的分类，计算 x 和 y 到新节点 z 的距离，即 x，y 的枝长：

$$d_{xz} = \frac{1}{2}(d_{xy} + d_x - d_y)$$

$$d_{yz} = \frac{1}{2}(d_{xy} + d_y - d_x)$$

   d) 计算新类 z 与其它类 u 的距离，更新距离矩阵：

$$d_{uz} = \frac{1}{2}(d_{xu} + d_{yu} - d_{xy})$$

   e) 重复步骤 2（更新其它类 u 的枝长，计算新增类 z 的枝长）
4. 对于最后三个节点(x, y, z)，每个节点的枝长为 $\frac{1}{2}(d_x - d_{yz})$

简化 NJ 算法：

使用 u 节点到所有其它节点之和D_u，代替其到星状树中心的距离d_u。

1. 将所有类初始化为一棵星状树，即每个分类单元成一类。
2. 对所有的分类单元，计算其到所有节点的距离之和为：

$$D_u = \sum_{i \neq u} d_{ui}$$

3. 如果有三个以上的分类存在，则继续执行循环：
   a) 选择一对分类单元 x 与 y，使 $d_{xy} - \frac{1}{n-2}(D_x + D_y)$ 最小
   b) 从星状树中删除 x 与 y，并将它们作为新类 z 的子类
   c) 变化后，x，y 的枝长为：

$$d'_x = \frac{1}{2}[d_{xy} + \frac{1}{n-2}(D_x - D_y)]$$

$$d'_y = \frac{1}{2}[d_{xy} - \frac{1}{n-2}(D_x - D_y)]$$

   d) 新加入类 z 到其它类 u 的距离之和

$$D'_z = \frac{1}{2} \times (d_x + d_y - n \cdot d_{xy})$$

   e) 计算新类 z 与其它类 u 的距离，以及 z 与 u 到其它节点的距离之和：

$$d_{uz} = \frac{1}{2}(d_{xu} + d_{yu} - d_{xy})$$

$$D'_u = D_u - d_{xu} - d_{yu} + d_{uz}$$

4. 对于最后三个节点(x, y, z)，每个节点的枝长为 $\frac{1}{2}(D_x - d_{yz})$。（n=3 时$D_x = d_x$）