**Neighbor Joining Algorithm:**

**Given:** An $n \times n$ distance matrix $D$

**Find:** Unrooted Phylogenetic $T$ with branch lengths. If $D$ is additive, then $d_T(i, j) = D[i, j]$ for all $1 \leq i, j \leq n$. Otherwise, $d_T(i, j) \approx D[i, j]$

**Terminology** Given $n \times n$ distance matrix $D$:
- Define $u_i = \sum_{k=1}^{n} D[C_i, C_k]$
- Define $S_D(C_i, C_j) = (n-2)D[C_i, C_j] - u_i - u_j$

**Algorithm Sketch**

**Initialization:**
- Form $n$ clusters $\{C_1, C_2, ..., C_n\}$, one for each species.
- Define tree $T$ to be the set of leaf nodes, one per species.

**Iteration:** ($D$ is currently $m \times m$)
- Pick $C_x, C_y = \arg\min_{i,j} S_D(C_i, C_j)$
- Merge $C_x$ and $C_y$ into new node $(C_x, C_y)$ in $T$.
- Assign length $\frac{1}{2}(D[C_x, C_y] + \frac{1}{(m-2)}(u_x - u_y))$ to edge $(C_x, (C_x, C_y))$
- Assign length $\frac{1}{2}(D[C_x, C_y] + \frac{1}{(m-2)}(u_y - u_x))$ to edge $(C_y, (C_x, C_y))$
- Remove rows and columns from $D$ corresponding to $C_x$ and $C_y$.
- Add row and column to $D$ for new vertex $(C_x, C_y)$.
- Set $D((C_x, C_y), C_z) = \frac{1}{2}(D[C_x, C_z] + D[C_y, C_z] - D[C_x, C_y])$ for all remaining clusters $C_z$.

**Termination:**
- When two clusters $C_x$ and $C_y$ remain, join them with an edge of length $D[C_x, C_y]$
**Practice:** Use the Neighbor Joining Algorithm to build the tree for the following distance matrix:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>