# Evolutionary Trees


Tattoos by Shauna S.

Black Tattoos

Color Tattoos

Art

About

Contact

FAQ

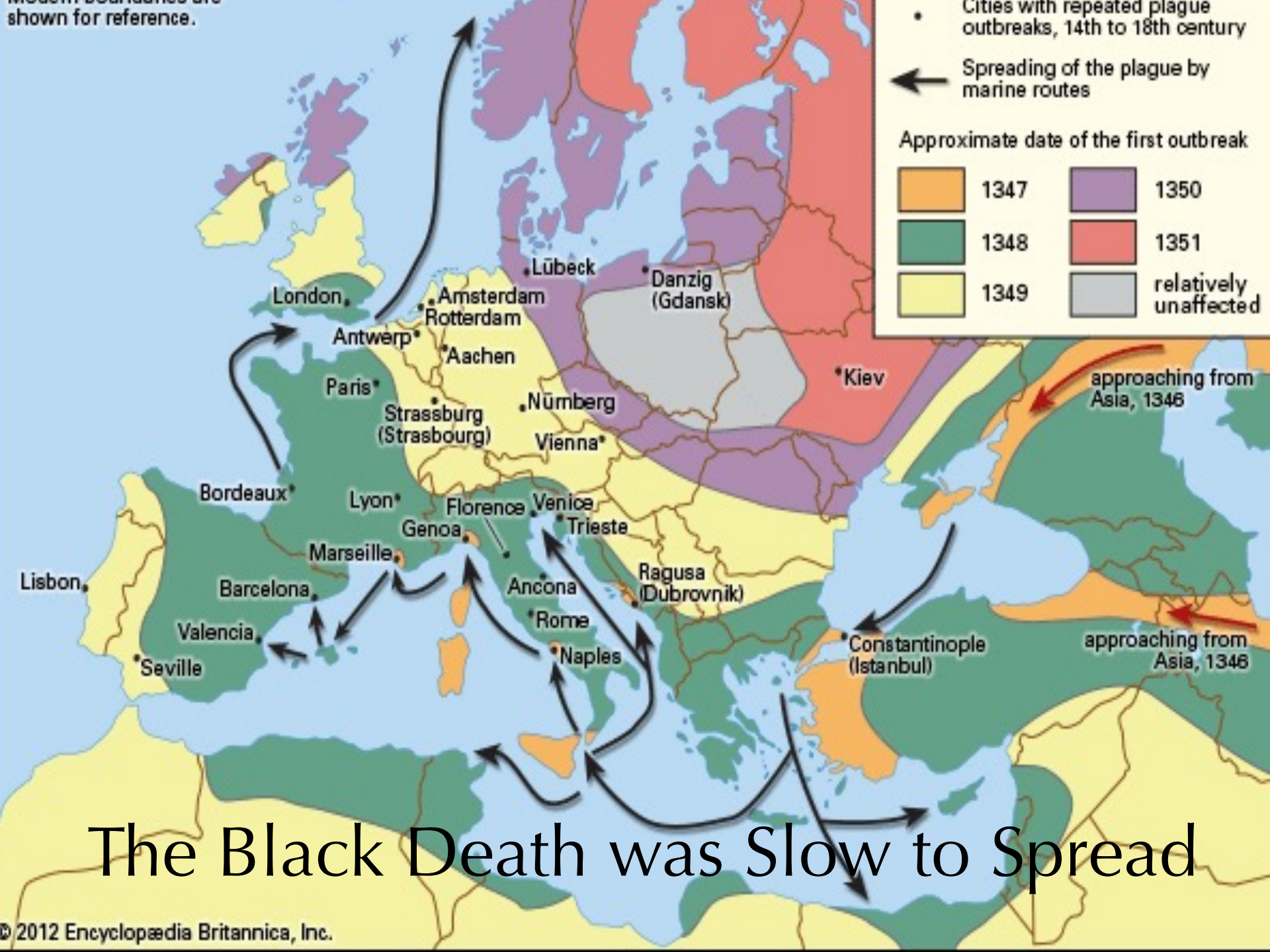
Health and Safety

Merchandise

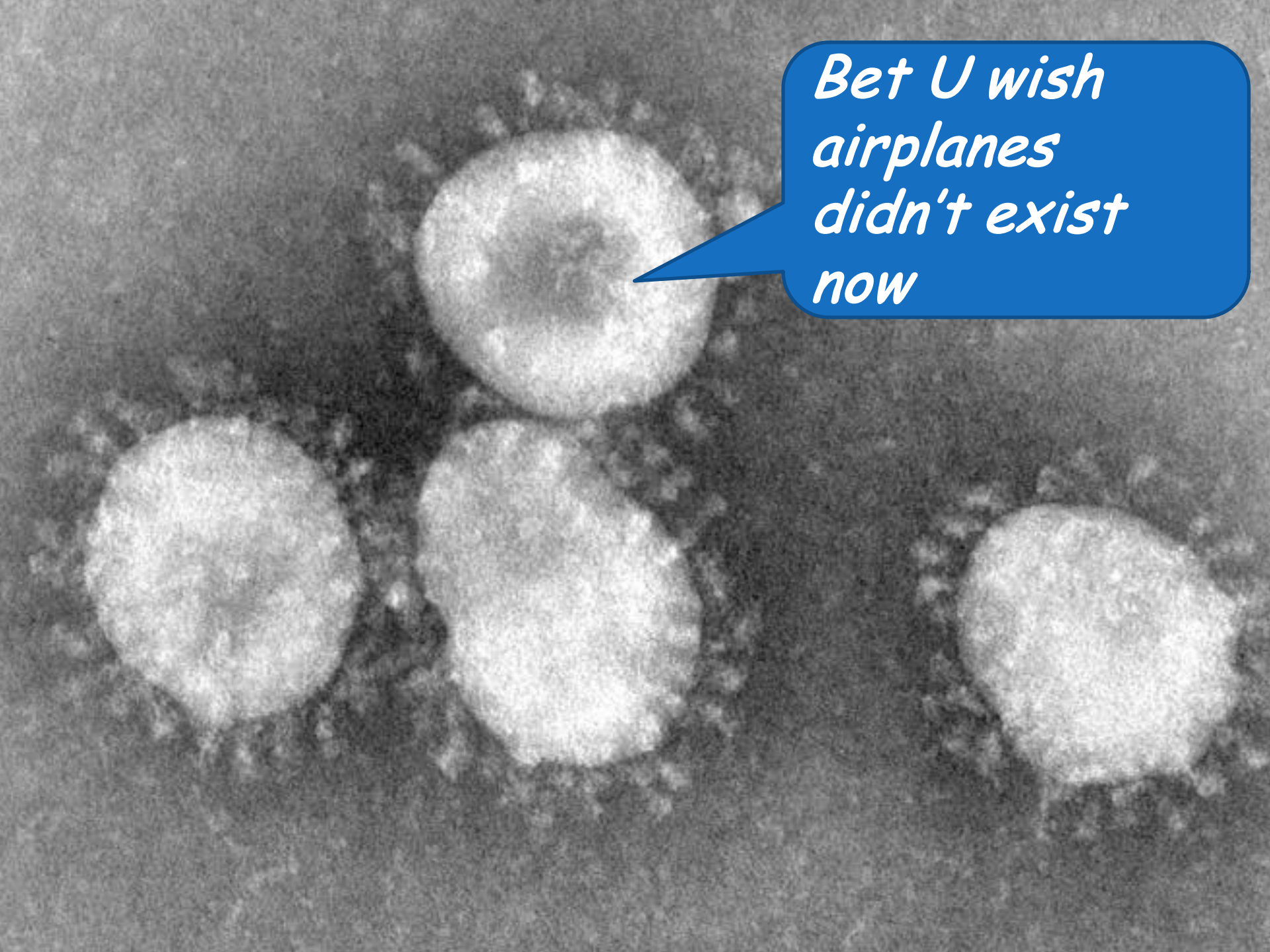
Art Commissions

Travel Calendar

Instagram




# The Black Death was Slow to Spread


Bet U wish airplanes didn't exist now

# Questions about ~~SARS~~ SARS-CoV-2

1. Which animal gave us SARS-CoV-2?
2. How does SARS-CoV-2 compare to other related viruses?
3. How has SARS-CoV-2 mutated over time?

# Questions about ~~SARS~~ SARS-CoV-2

1. Which animal gave us SARS-CoV-2?
2. How does SARS-CoV-2 compare to other related viruses?
3. How has SARS-CoV-2 mutated over time?

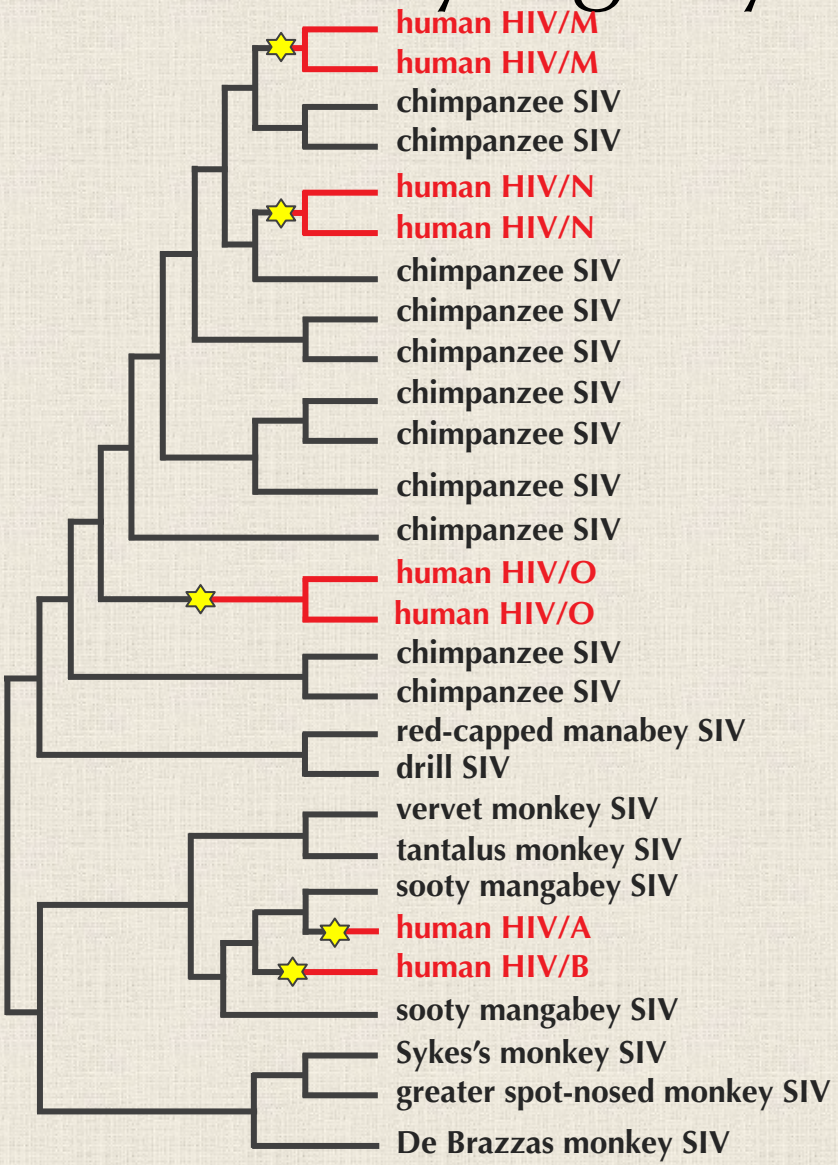
All of these questions require us to construct **evolutionary trees** (a.k.a. **phylogenies**).

# Another Example: HIV Phylogeny

SIVs (monkeys)

HIV (human)

human infection



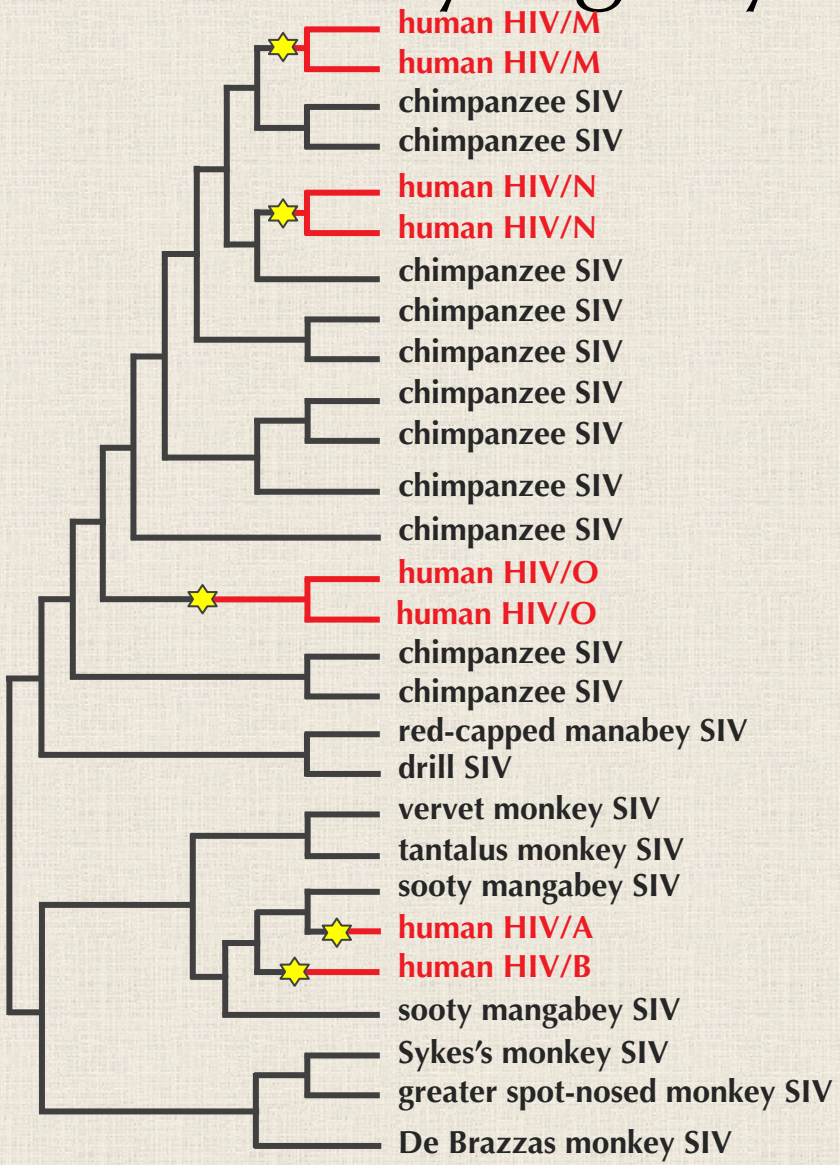
# Another Example: HIV Phylogeny

SIVs (monkeys)

HIV (human)

human infection

**STOP:** What conclusion can we draw from this tree?



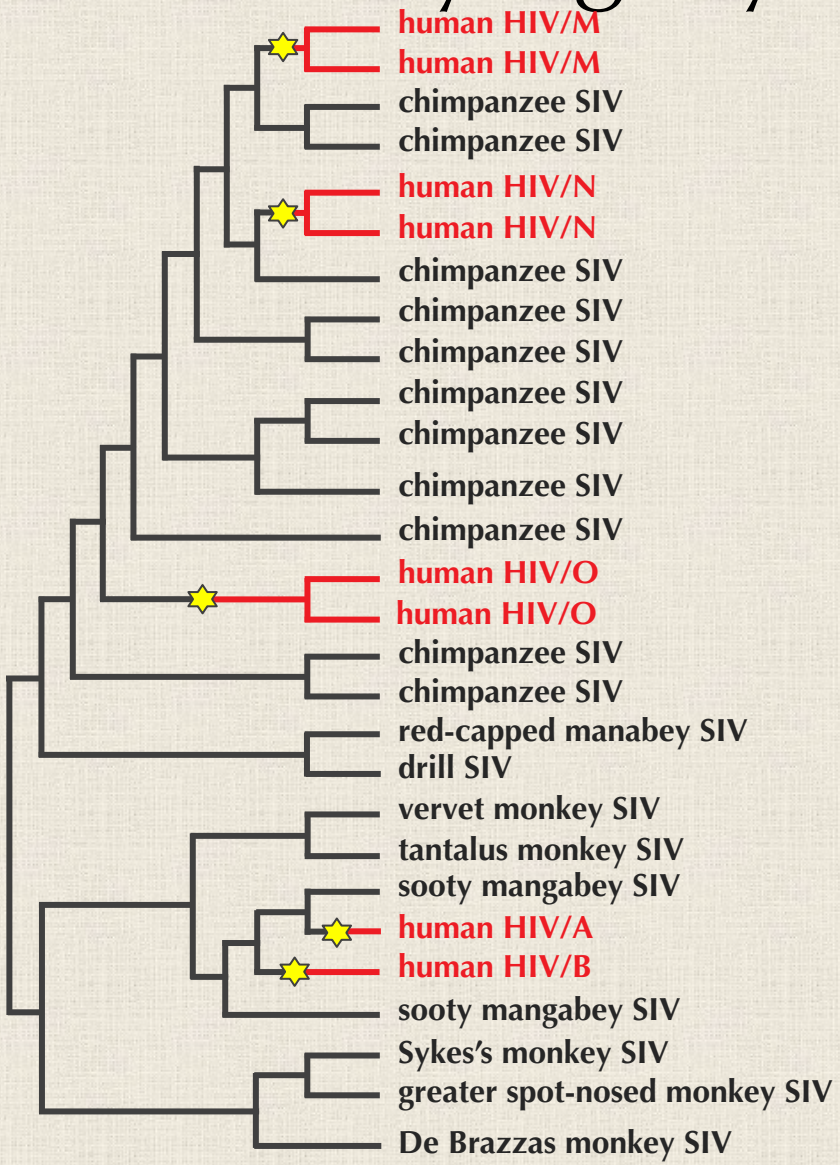
# Another Example: HIV Phylogeny

SIVs (monkeys)

HIV (human)

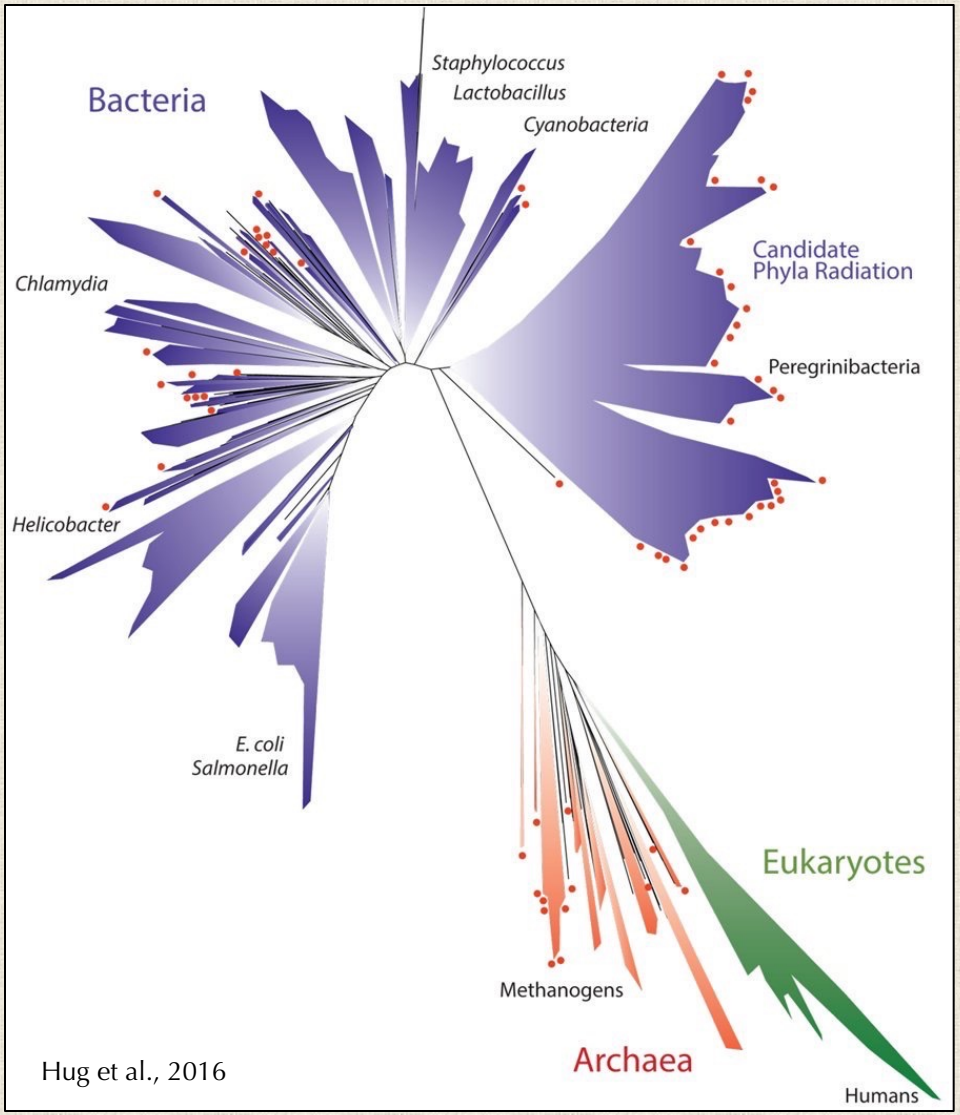
human infection

**Answer:** HIV comes from *multiple* origins in chimps, not one.



# Two Computational Questions

How do we construct the tree's *structure?*

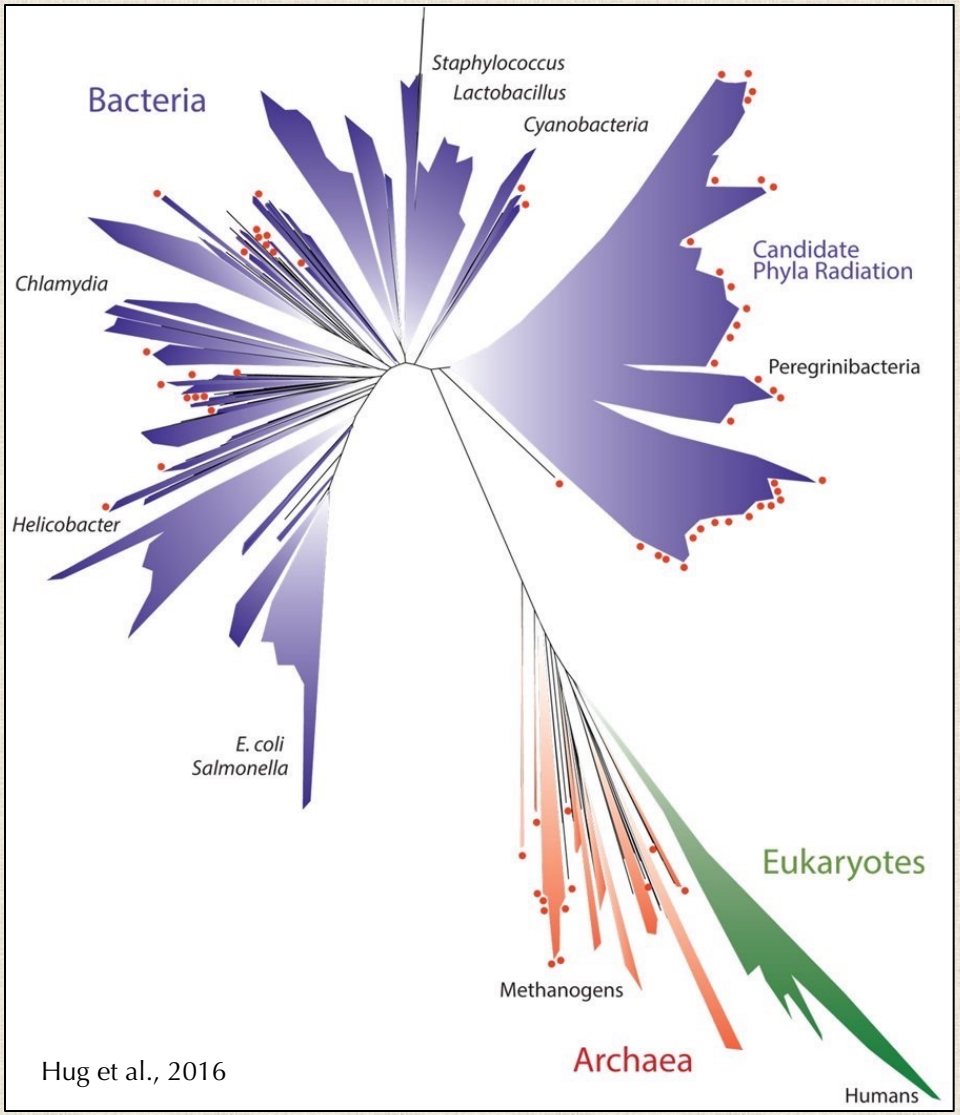



Hug et al., 2016

# Two Computational Questions

How do we construct the tree's *structure?*

Can we infer anything about the ancestral species on the inside of the tree?




Hug et al., 2016

# Some Tree Math


**Tree:** Connected graph containing no cycles.

# Some Tree Math


**Tree:** Connected graph containing no cycles.

**Leaves** (degree = 1): present-day species

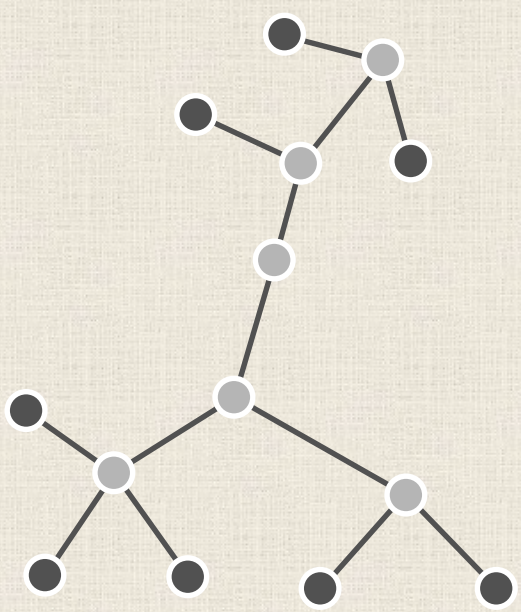
# Some Tree Math

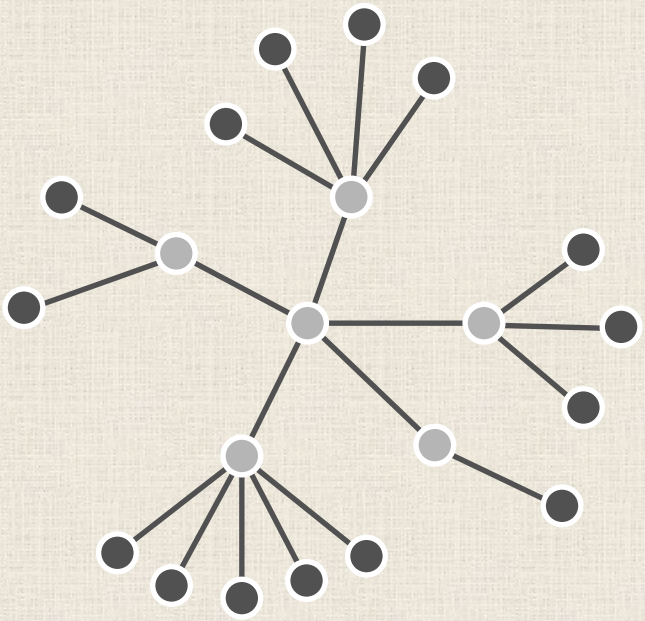

**Tree:** Connected graph containing no cycles.

**Leaves** (degree = 1): present-day species

**Internal nodes** (degree ≥ 2): ancestral species

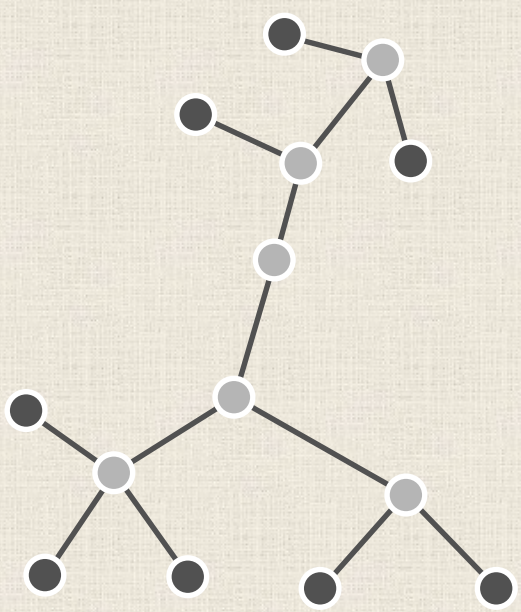
# Some Tree Math

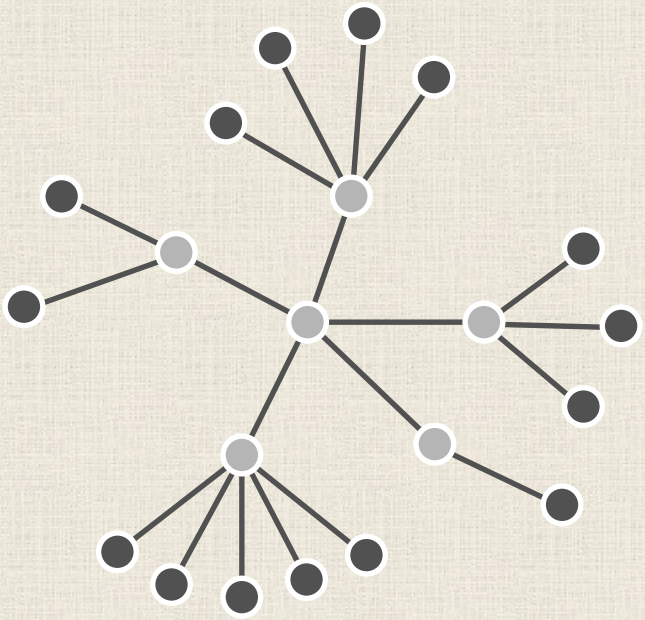




**Theorem:** Every tree with $n$ nodes has exactly $n - 1$ edges.

# Some Tree Math





**Theorem:** There is a unique path connecting any two nodes in a tree.


Courtesy: Universal Studios

# Character-Based Construction

Biologists used to form trees based on anatomical or physiological properties called **characters**.

# Character-Based Construction

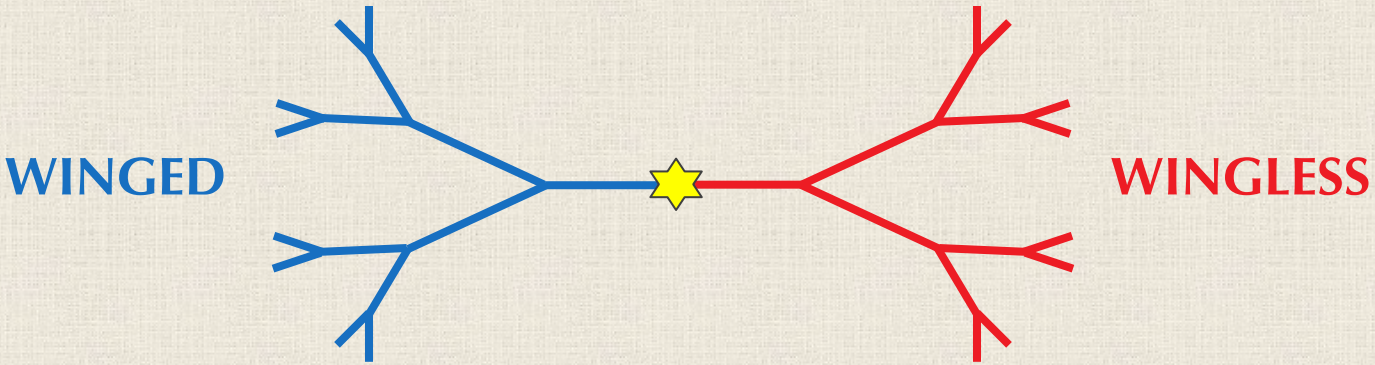
Biologists used to form trees based on anatomical or physiological properties called **characters**.

**STOP:** Say you wanted to construct an evolutionary tree of all insects. What might be the first thing you would do?

# Character-Based Construction

Biologists used to form trees based on anatomical or physiological properties called **characters**.

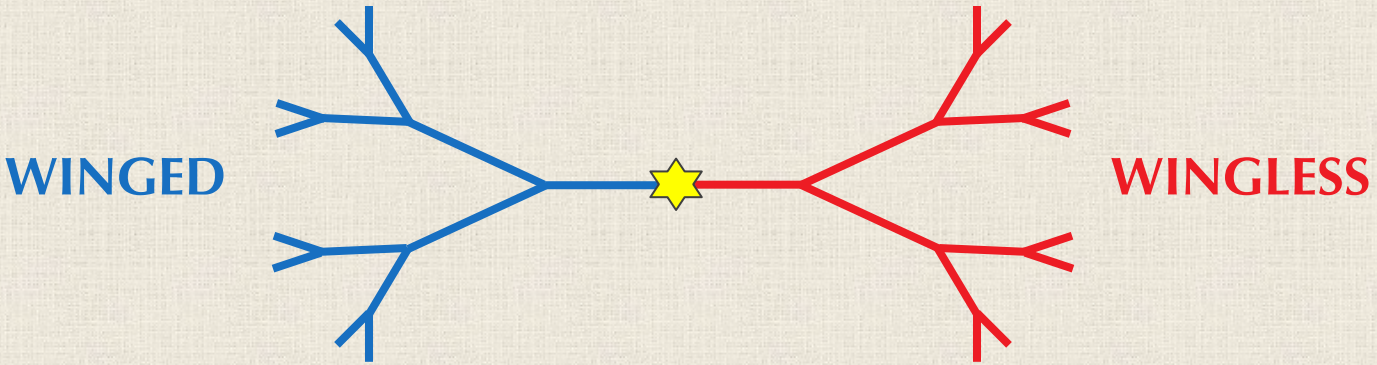
**STOP:** Say you wanted to construct an evolutionary tree of all insects. What might be the first thing you would do?



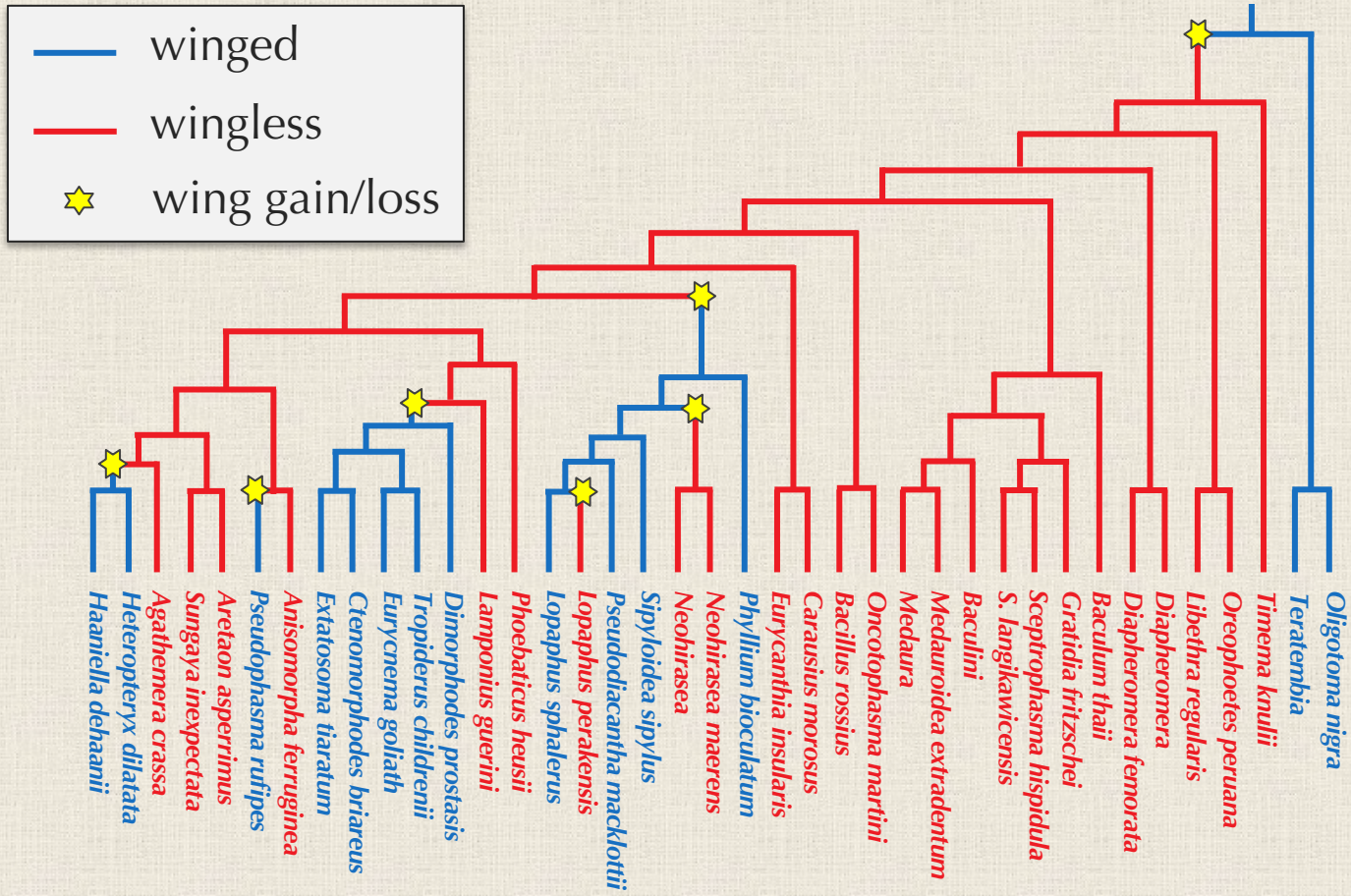
# Character-Based Construction

**Dollo's principle of irreversibility** (1893): evolution doesn't reinvent the same organ (e.g. insect wings).

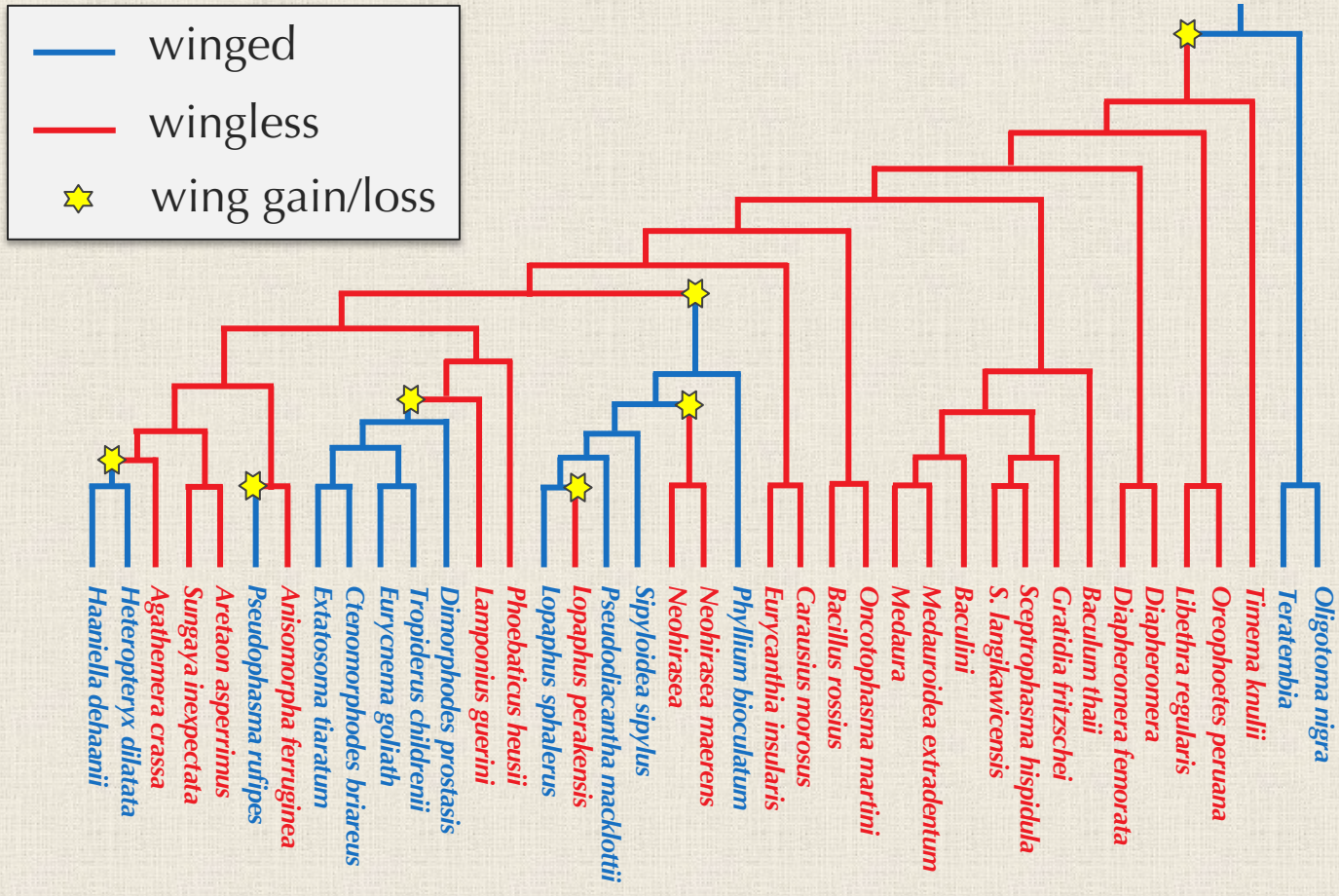
**STOP:** Say you wanted to construct an evolutionary tree of all insects. What might be the first thing you would do?



# Dollo's Principle Violated in Stick Insect Phylogeny

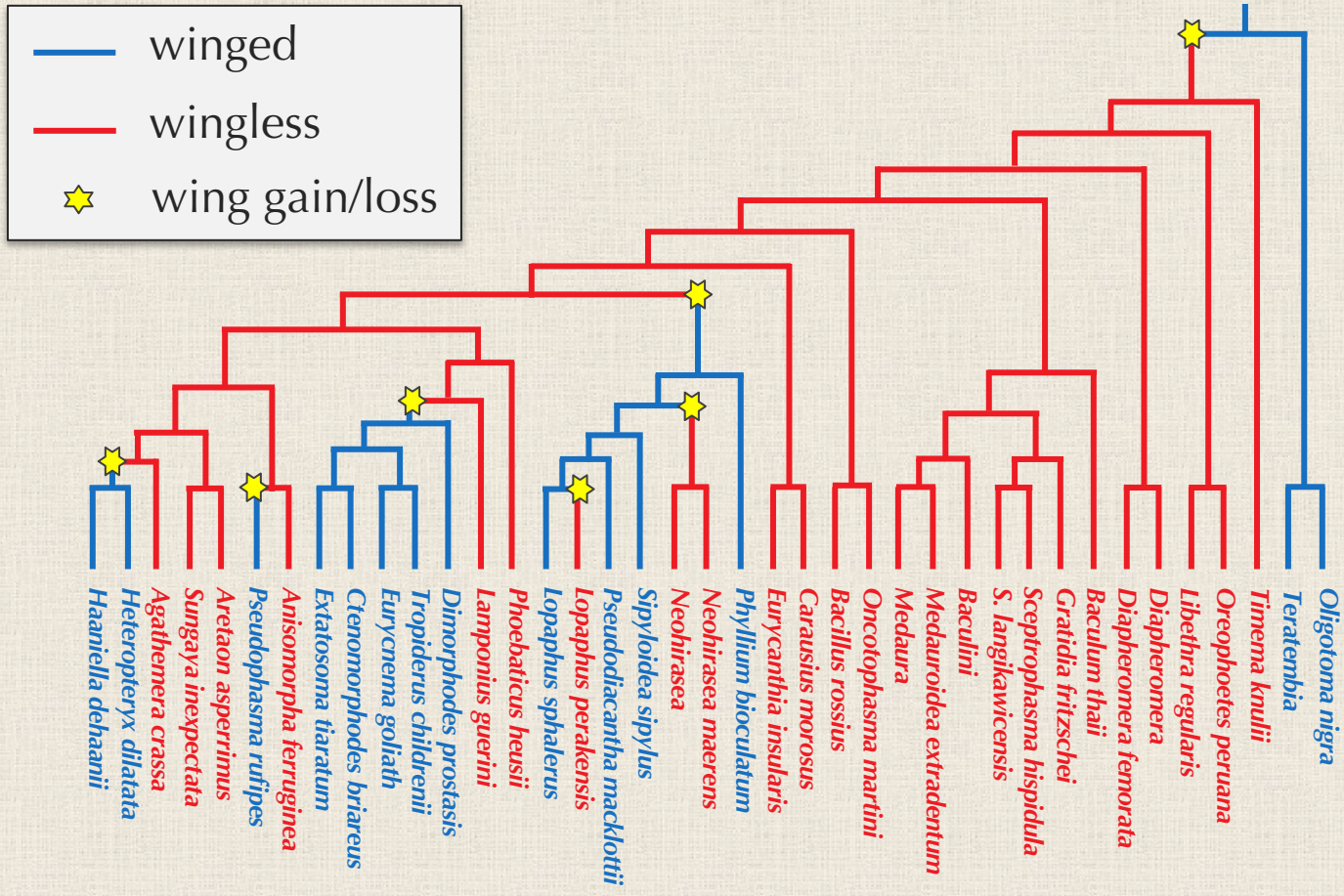


# Dollo's Principle Violated in Stick Insect Phylogeny




**STOP (biologists):** What do you think happened?

# Dollo's Principle Violated in Stick Insect Phylogeny



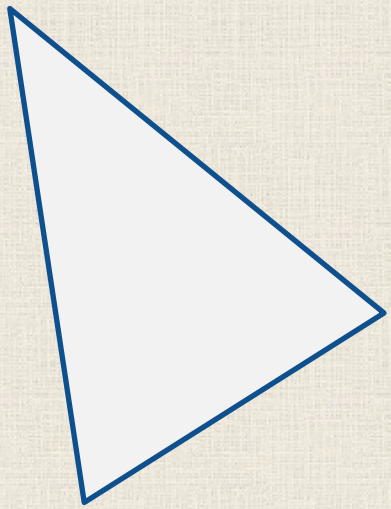

**Key Point:** We need a more rigorous approach.

# TRANSFORMING DISTANCE MATRICES INTO TREES

# Definition of a Distance Matrix

**Distance matrix:** A matrix $D$ representing distances between pairs of $n$ organisms that satisfies three properties:

1. **Symmetry:** $D_{i,j} = D_{j,i}$ for all pairs $i$, $j$
2. **Non-negativity:** $D_{i,j} >= 0$ for all pairs $i$, $j$
3. **Triangle inequality:** For all $i$, $j$, and $k$, $D_{i,j} + D_{j,k} >= D_{i,k}$ .



# A Multiple Alignment Defines a Simple Distance Matrix

| Species | Alignment |
|---|---|
| **Chimp** | ACGTAGGCCT |
| **Human** | ATGTAAGACT |
| **Seal** | TCGAGAGCAC |
| **Whale** | TCGAAAGCAT |

# A Multiple Alignment Defines a Simple Distance Matrix

$D_{i,j}$ = number of differing symbols between $i$-th and $j$-th rows of a multiple alignment.

| Species | Alignment | Distance Matrix | | | |
|---|---|---|---|---|---|
| | | **Chimp** | **Human** | **Seal** | **Whale** |
| **Chimp** | ACGTAGGCCT | 0 | 3 | 6 | 4 |
| **Human** | ATGTAAGACT | 3 | 0 | 7 | 5 |
| **Seal** | TCGAGAGCAC | 6 | 7 | 0 | 2 |
| **Whale** | TCGAAAGCAT | 4 | 5 | 2 | 0 |

# A Multiple Alignment Defines a Simple Distance Matrix

$D_{i,j}$ = number of differing symbols between $i$-th and $j$-th rows of a multiple alignment.

| Species | Alignment | Distance Matrix | | | |
|---|---|---|---|---|---|
| | | **Chimp** | **Human** | **Seal** | **Whale** |
| **Chimp** | ACGTAGGCCT | 0 | **3** | 6 | 4 |
| **Human** | ATGTAAGACT | **3** | 0 | 7 | 5 |
| **Seal** | TCGAGAGCAC | 6 | 7 | 0 | 2 |
| **Whale** | TCGAAAGCAT | 4 | 5 | 2 | 0 |

# Distance-Based Phylogeny

**Distance-Based Phylogeny Problem**.

- **Input:** A distance matrix.
- **Output:** The tree fitting this distance matrix.

# Distance-Based Phylogeny

**Distance-Based Phylogeny Problem**.

- **Input:** A distance matrix.
- **Output:** The tree *fitting* this distance matrix.

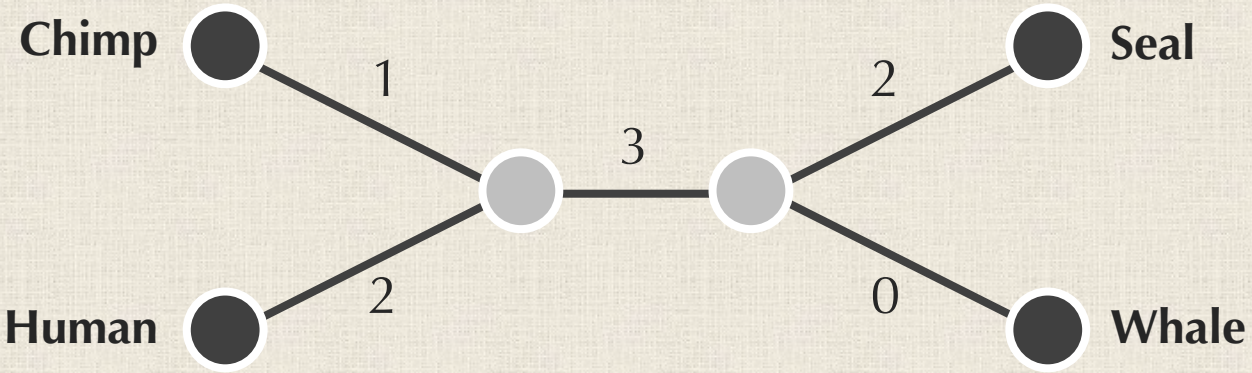
Of course, we are getting a bit ahead of ourselves – we should define what we mean by "fitting"!

# “Fitting” a Tree to a Matrix

| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |

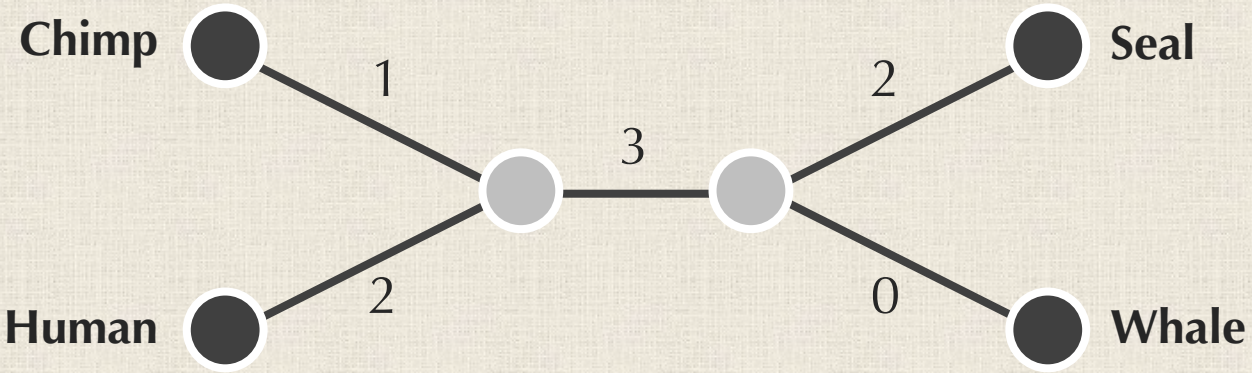
# “Fitting” a Tree to a Matrix

| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |



# "Fitting" a Tree to a Matrix

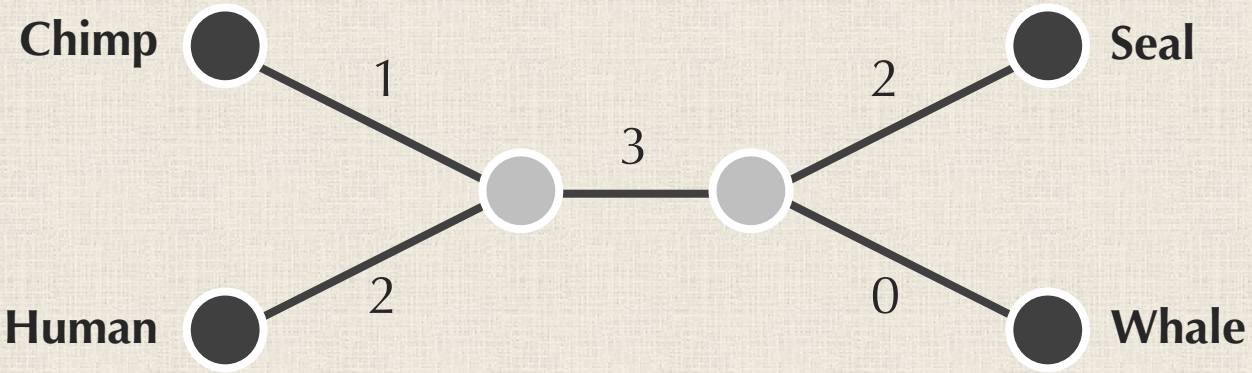
| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |




$d_{i,j}(T)$ = distance between nodes $i$ and $j$ in tree $T$, computed by summing edge weights from $i$ to $j$.

# "Fitting" a Tree to a Matrix

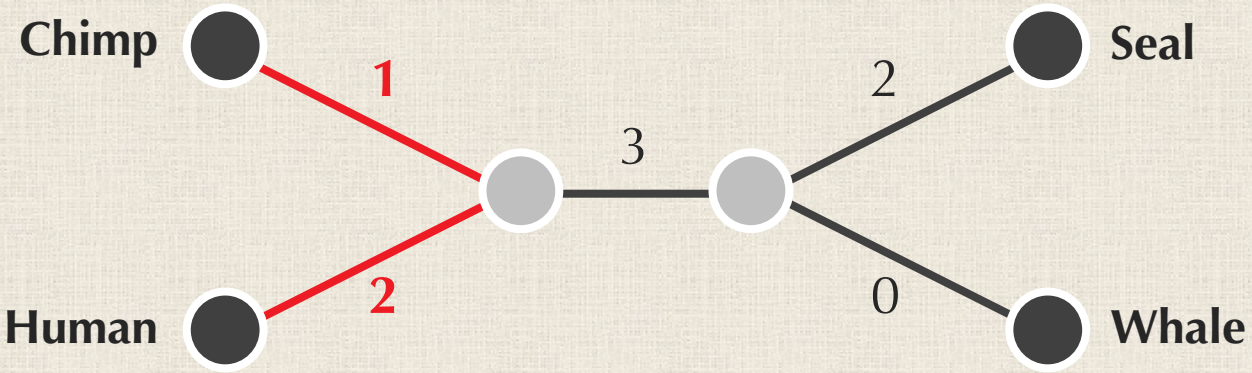
| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |




We say that $T$ **fits** matrix $D$ if for every pair $i$ and $j$, $d_{i,j}(T) = D_{i,j}$.

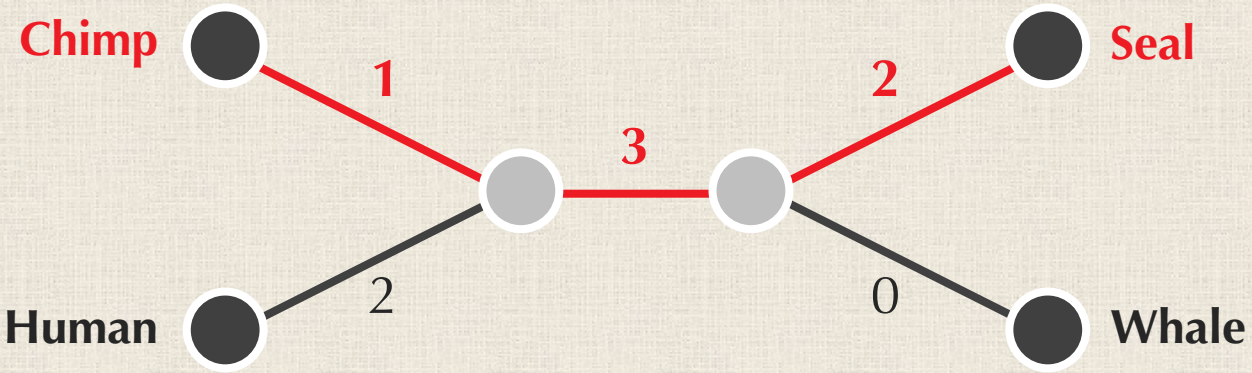
# “Fitting” a Tree to a Matrix

| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | **3** | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |



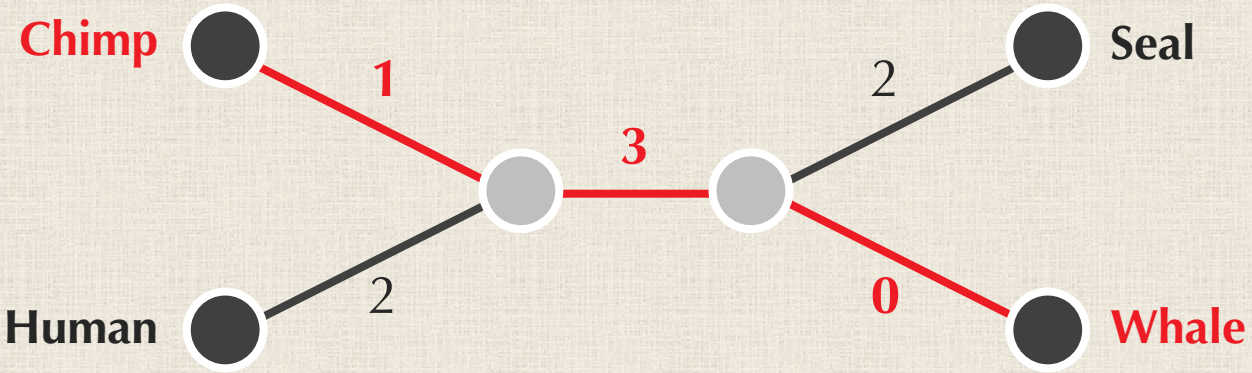
# “Fitting” a Tree to a Matrix

| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | **6** | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |



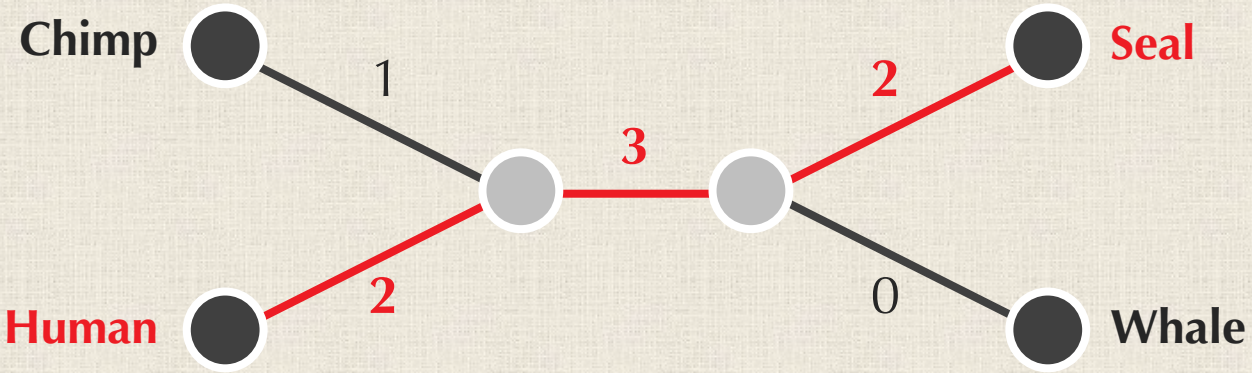
# “Fitting” a Tree to a Matrix

| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | **4** |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |



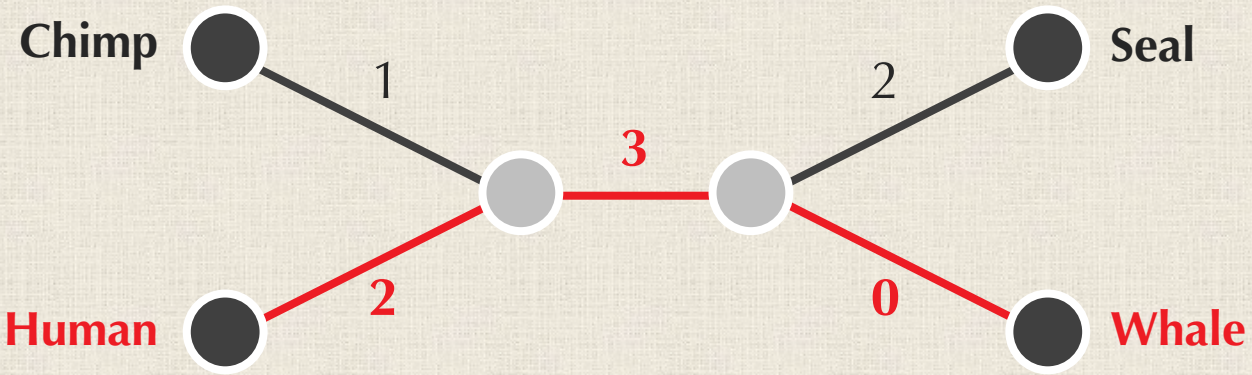
# “Fitting” a Tree to a Matrix

| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | **7** | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |



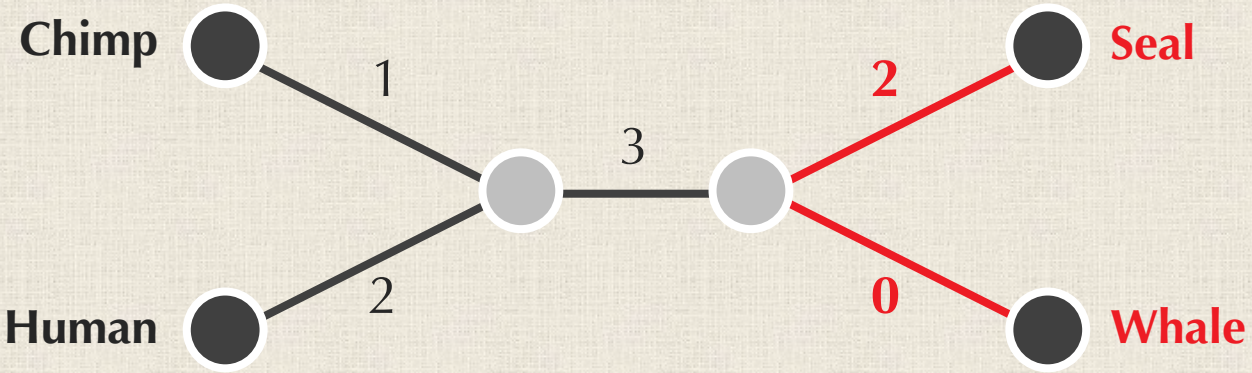
# “Fitting” a Tree to a Matrix

| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | **5** |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |



# “Fitting” a Tree to a Matrix

| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | **2** |
| **Whale** | 4 | 5 | 2 | 0 |



# Return to Distance-Based Phylogeny

**Exercise:** Find a tree fitting the following matrix.

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 3 | 4 | 3 |
| $v_2$ | 3 | 0 | 4 | 5 |
| $v_3$ | 4 | 4 | 0 | 2 |
| $v_4$ | 3 | 5 | 2 | 0 |

# Sometimes, **No** Tree Fits a Matrix

There is no tree fitting this matrix!

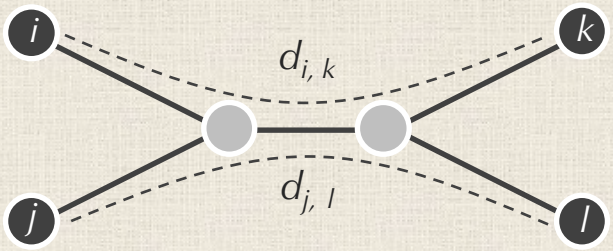
| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 3 | 4 | 3 |
| $v_2$ | 3 | 0 | 4 | 5 |
| $v_3$ | 4 | 4 | 0 | 2 |
| $v_4$ | 3 | 5 | 2 | 0 |

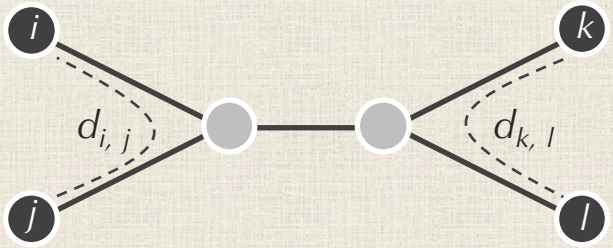
**Additive matrix:** distance matrix such that there exists a tree fitting it.

# How Can We Know if a Matrix Has a Tree Fitting It?

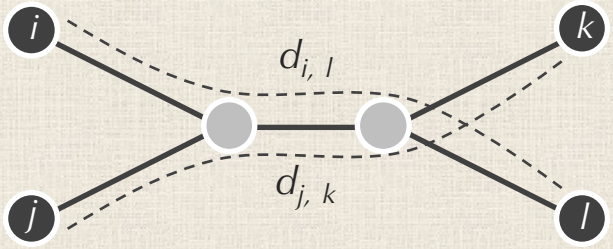
Indices ($i$, $j$, $k$, $l$) satisfy the **four point condition** if two of the following sums are equal, and the third sum is less than or equal to the other two sums:

$$D_{i,j} + D_{k,l} \quad D_{i,k} + D_{j,l} \quad D_{i,l} + D_{j,k}$$



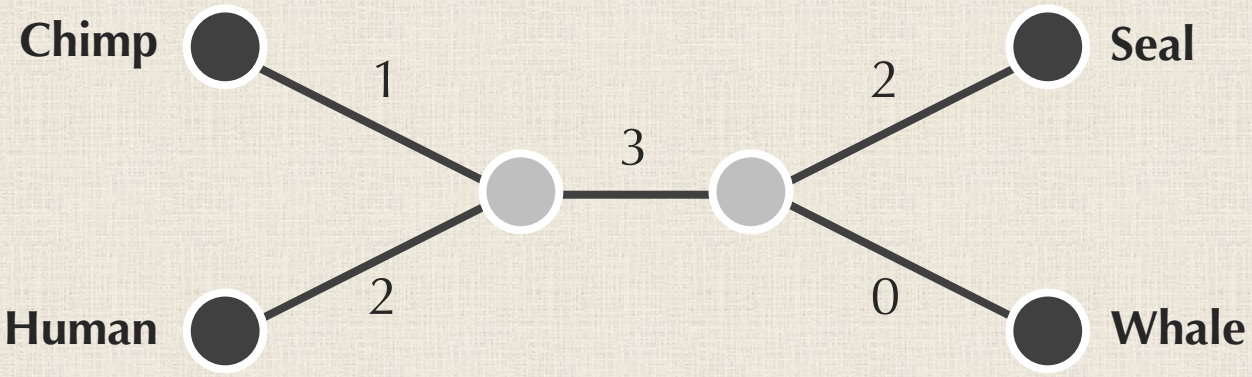




**Four Point Theorem:** A distance matrix $D$ is additive if and only if the four point condition holds for every quartet ($i$, $j$, $k$, $l$).



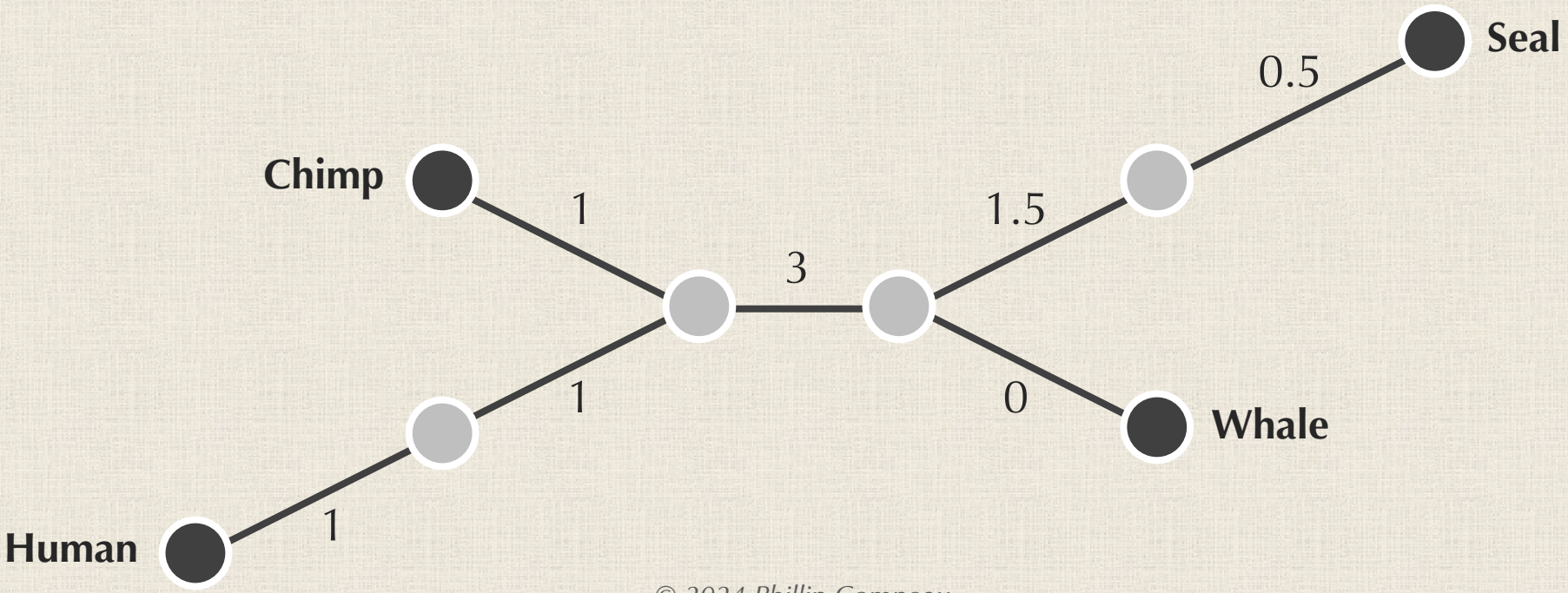
# Sometimes, **More Than One** Tree Fits a Matrix

| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |

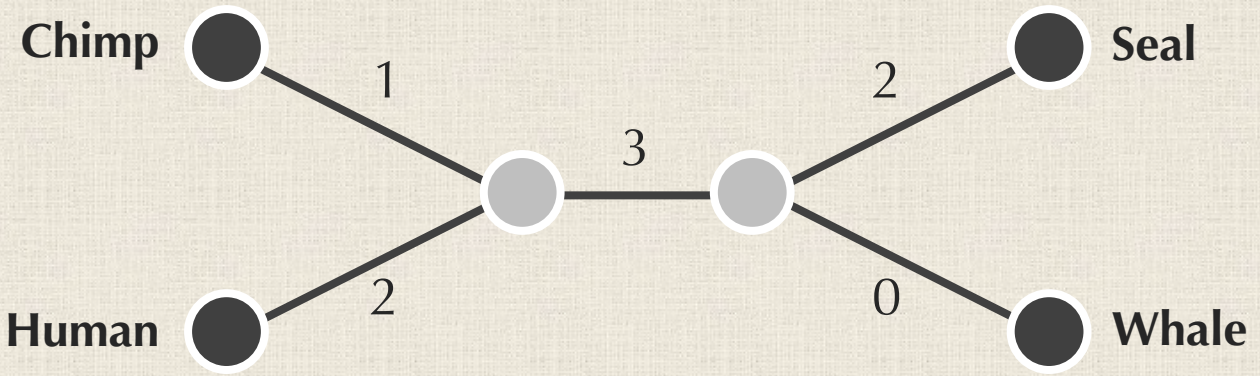


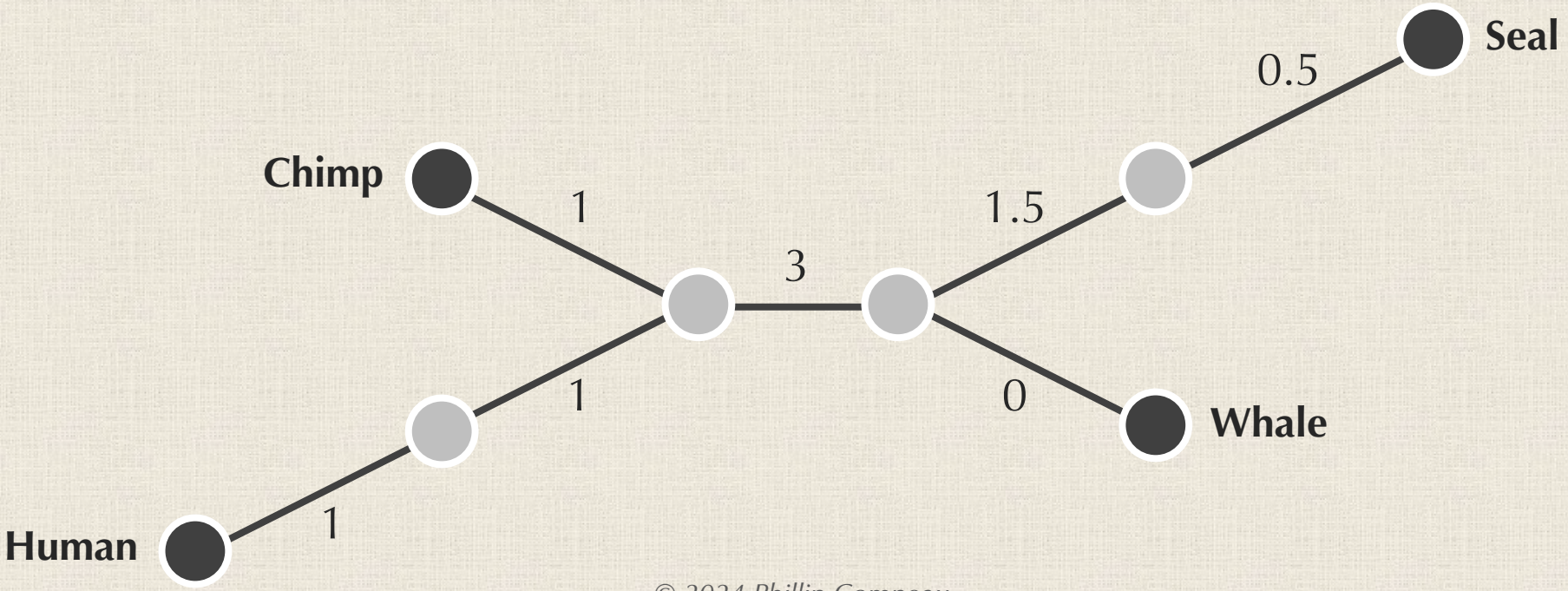
# Sometimes, **More Than One** Tree Fits a Matrix

| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |

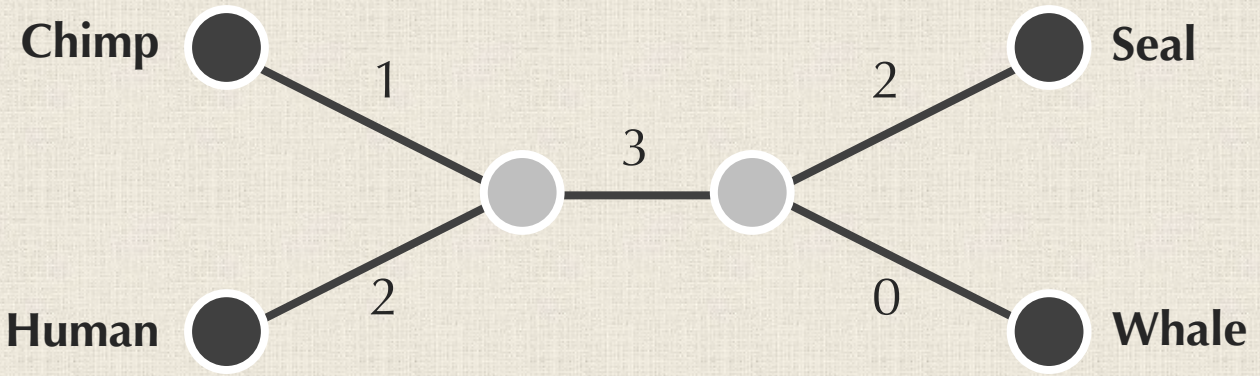


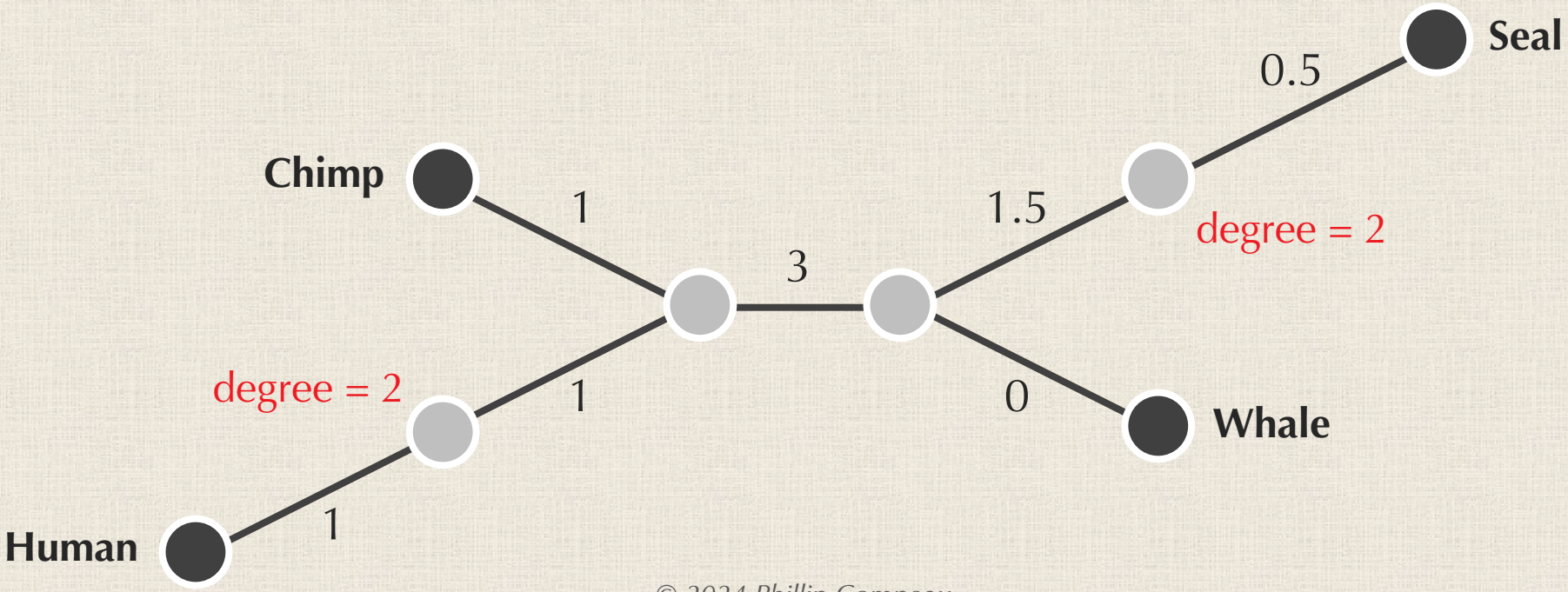
# Which Tree is Better?



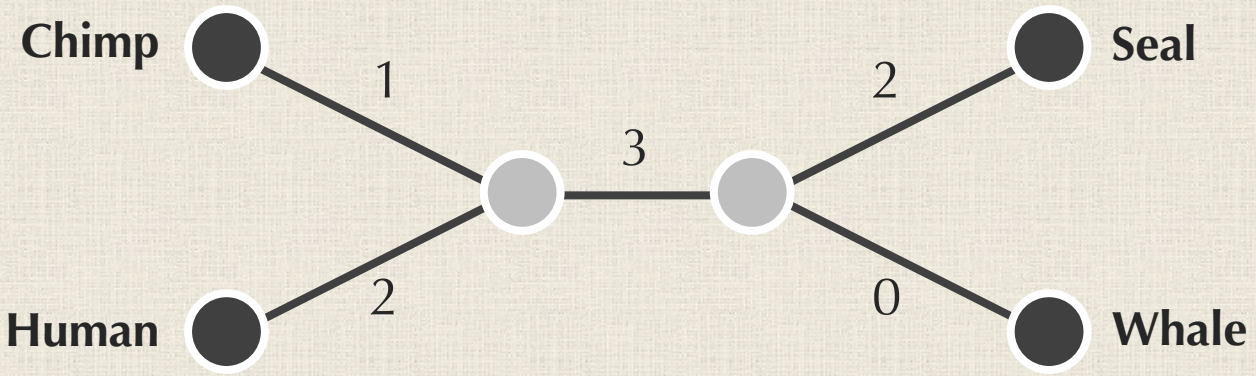



# Which Tree is Better?



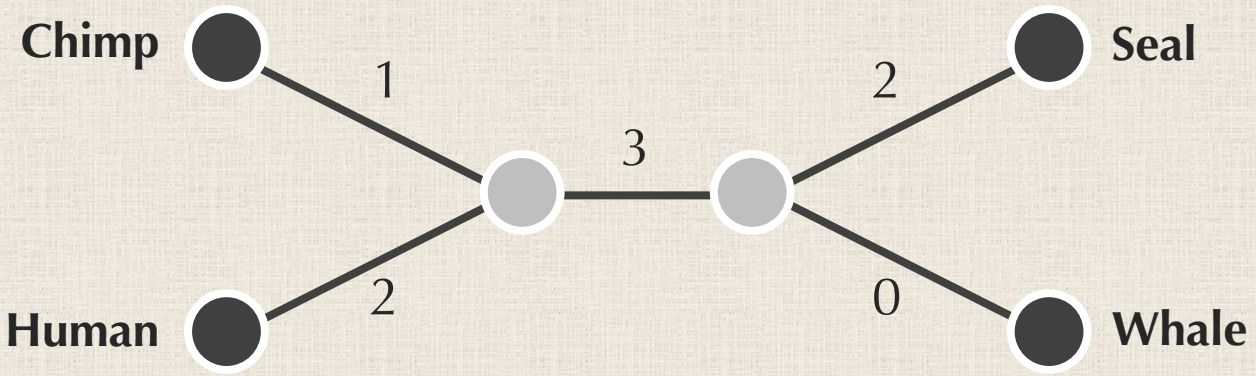



# Which Tree is Better?




**Simple tree:** tree with no nodes of degree 2.

# Which Tree is Better?




**Simple tree:** tree with no nodes of degree 2.

**Theorem:** There is a unique *simple* tree fitting an *additive* matrix. (And there is an algorithm to produce this tree.)

# Reformulating Distance-Based Phylogeny

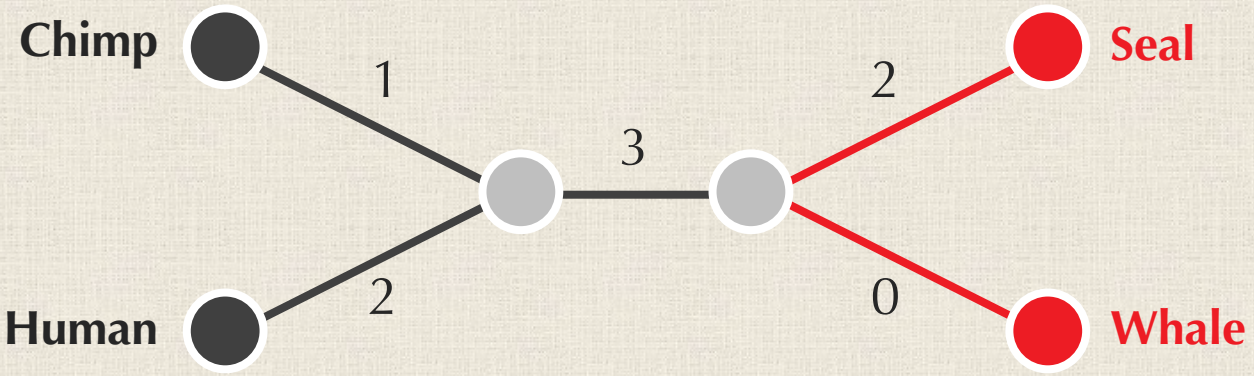
**Distance-Based Phylogeny Problem**: *Construct an evolutionary tree from a distance matrix.*

- **Input:** A distance matrix.
- **Output:** The simple tree fitting this distance matrix (if this matrix is additive).

# TOWARD AN ALGORITHM FOR DISTANCE-BASED PHYLOGENY

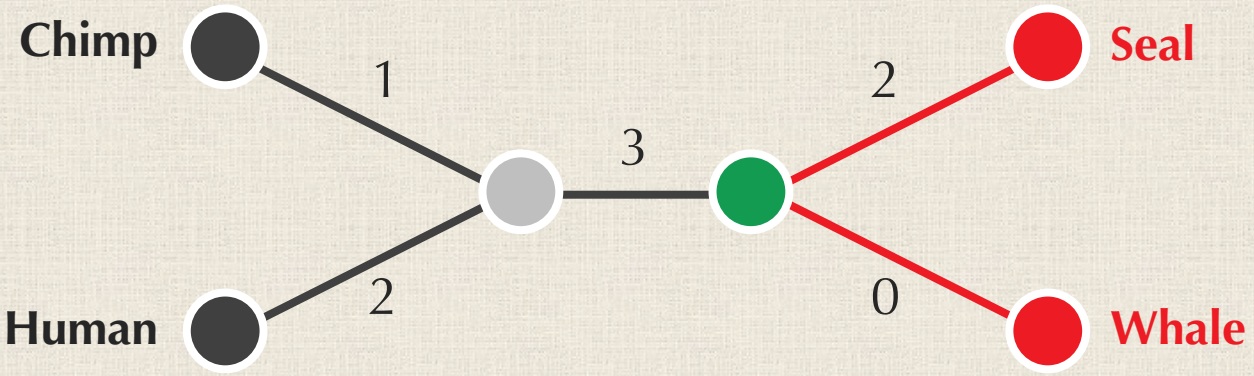
# An Idea for Distance-Based Phylogeny

| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | **2** |
| **Whale** | 4 | 5 | 2 | 0 |



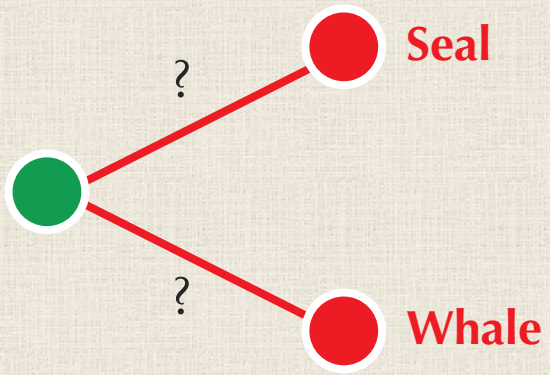
# An Idea for Distance-Based Phylogeny

Seal and whale are **neighbors** (meaning they are leaves with the same **parent**).



# An Idea for Distance-Based Phylogeny

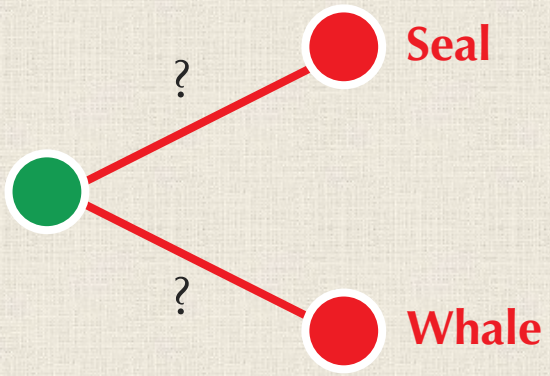
| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | **2** |
| **Whale** | 4 | 5 | 2 | 0 |



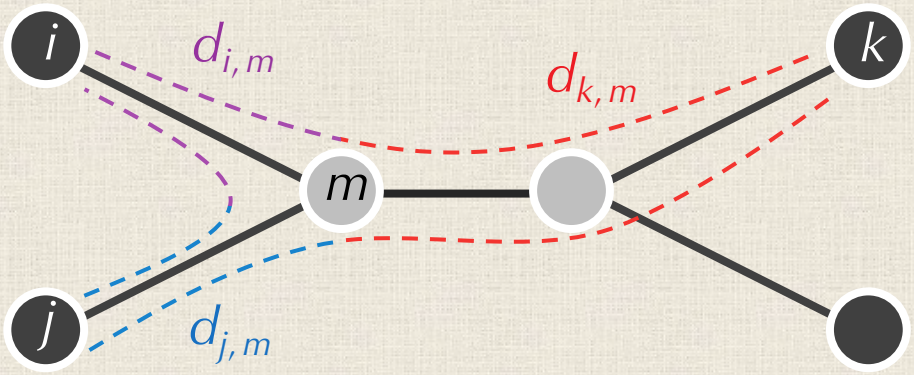
# An Idea for Distance-Based Phylogeny

| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | **2** |
| **Whale** | 4 | 5 | 2 | 0 |

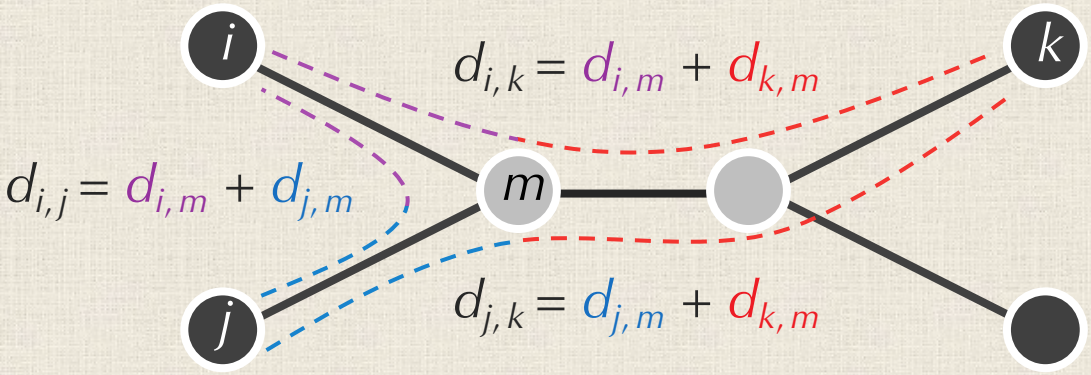
**Key Point:** How do we compute the unknown distances?



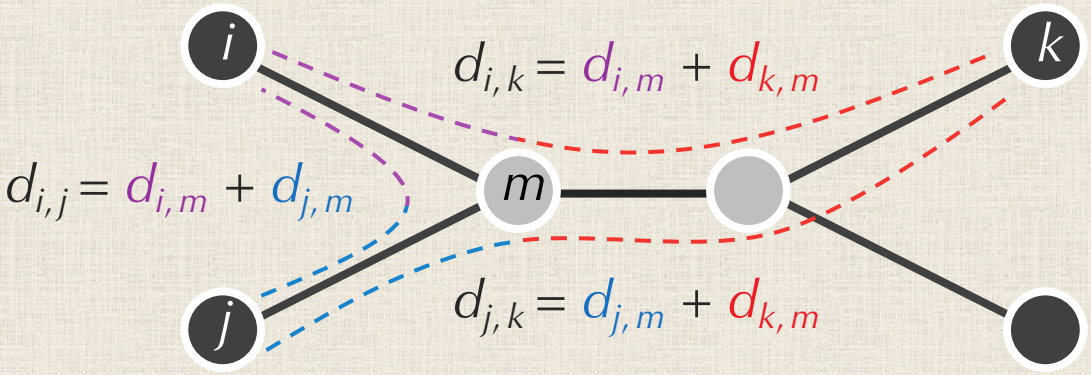
# Toward a Recursive Algorithm



# Toward a Recursive Algorithm

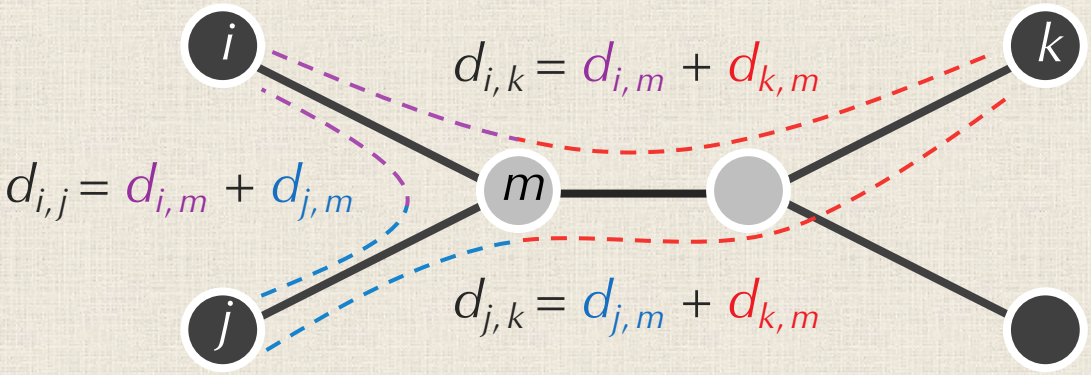


# Toward a Recursive Algorithm




$$d_{k,m} = [(d_{i,m} + d_{k,m}) + (d_{j,m} + d_{k,m}) - (d_{i,m} + d_{j,m})] / 2$$

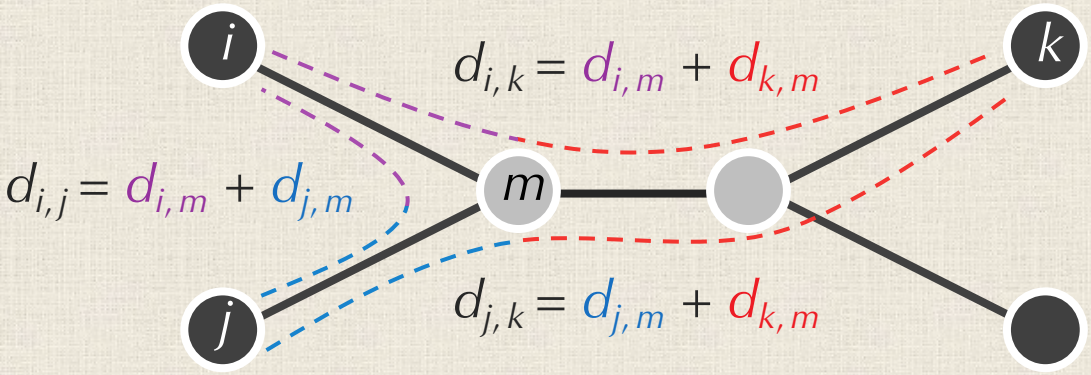
# Toward a Recursive Algorithm




$$d_{k,m} = [(d_{i,m} + d_{k,m}) + (d_{j,m} + d_{k,m}) - (d_{i,m} + d_{j,m})] / 2$$

$$d_{k,m} = (d_{i,k} + d_{j,k} - d_{i,j}) / 2$$

# Toward a Recursive Algorithm

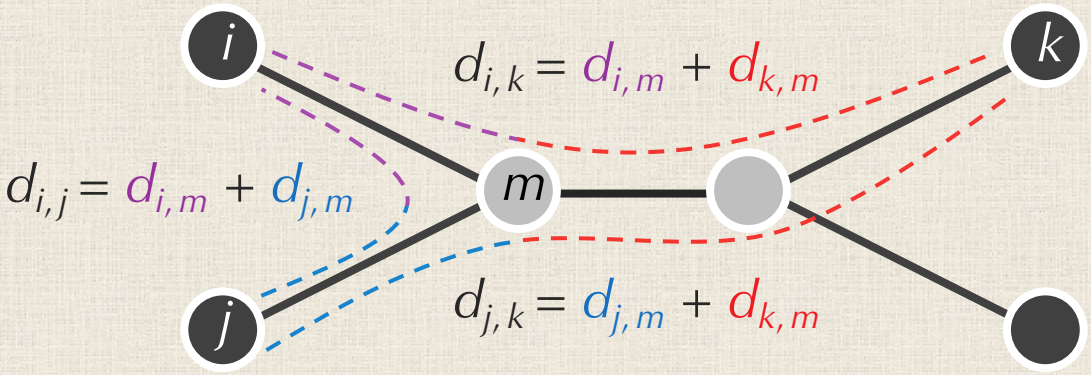



$$d_{k,m} = [(d_{i,m} + d_{k,m}) + (d_{j,m} + d_{k,m}) - (d_{i,m} + d_{j,m})] \,/\, 2$$

$$d_{k,m} = (d_{i,k} + d_{j,k} - d_{i,j}) \,/\, 2$$

$$d_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j}) \,/\, 2$$

# Toward a Recursive Algorithm



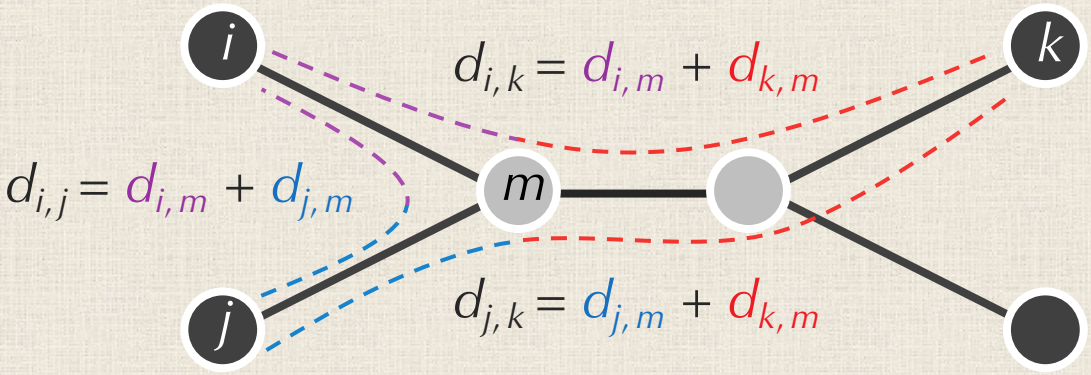

$$d_{k,m} = [(d_{i,m} + d_{k,m}) + (d_{j,m} + d_{k,m}) - (d_{i,m} + d_{j,m})] \,/\, 2$$

$$d_{k,m} = (d_{i,k} + d_{j,k} - d_{i,j}) \,/\, 2$$

$$d_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j}) \,/\, 2$$

$$\therefore \quad d_{i,m} = D_{i,k} - (D_{i,k} + D_{j,k} - D_{i,j}) \,/\, 2$$

# Toward a Recursive Algorithm




$$d_{k,m} = [(d_{i,m} + d_{k,m}) + (d_{j,m} + d_{k,m}) - (d_{i,m} + d_{j,m})] / 2$$

$$d_{k,m} = (d_{i,k} + d_{j,k} - d_{i,j}) / 2$$

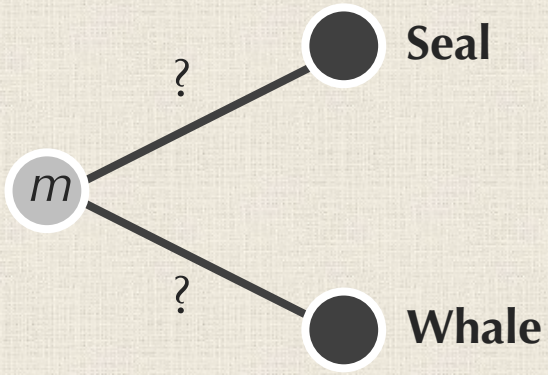
$$d_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j}) / 2$$

$$\therefore \quad d_{i,m} = D_{i,k} - (D_{i,k} + D_{j,k} - D_{i,j}) / 2$$

$$d_{i,m} = (D_{i,k} + D_{i,j} - D_{j,k}) / 2$$

# An Idea for Distance-Based Phylogeny

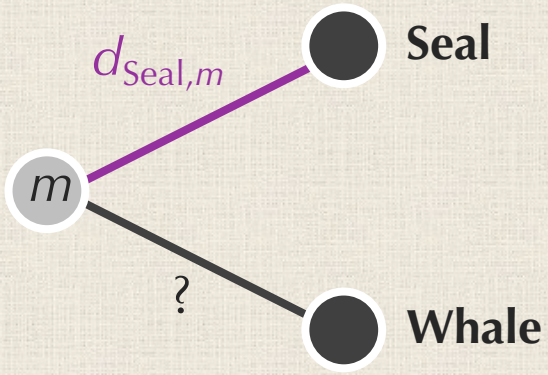
| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |




$$d_{i,m} = (D_{i,k} + D_{i,j} - D_{j,k}) / 2$$

# An Idea for Distance-Based Phylogeny

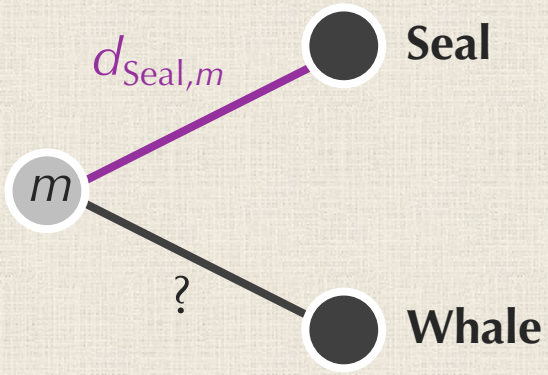
| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |




$$d_{Seal,m} = (D_{Seal,k} + D_{Seal,j} - D_{j,k}) / 2$$

# An Idea for Distance-Based Phylogeny

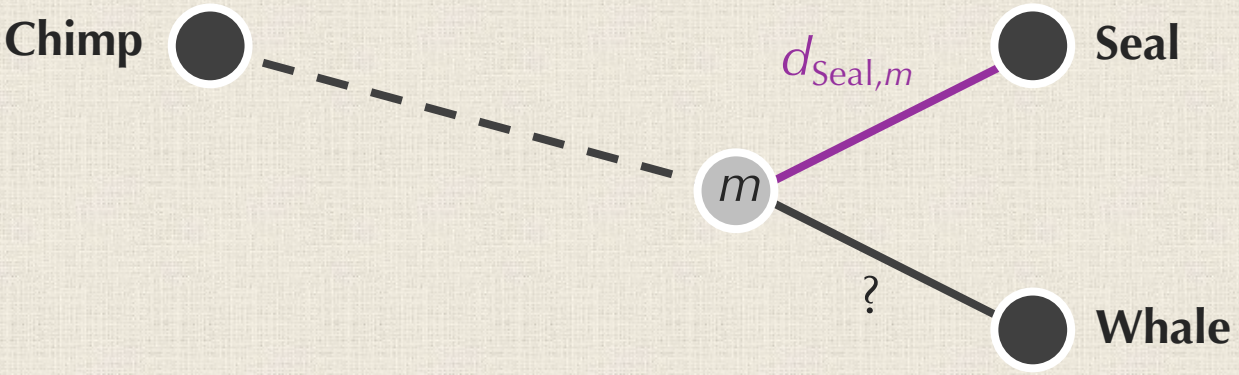
| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |




$$d_{Seal,m} = (D_{Seal,k} + D_{Seal,Whale} - D_{Whale,k}) / 2$$

# An Idea for Distance-Based Phylogeny

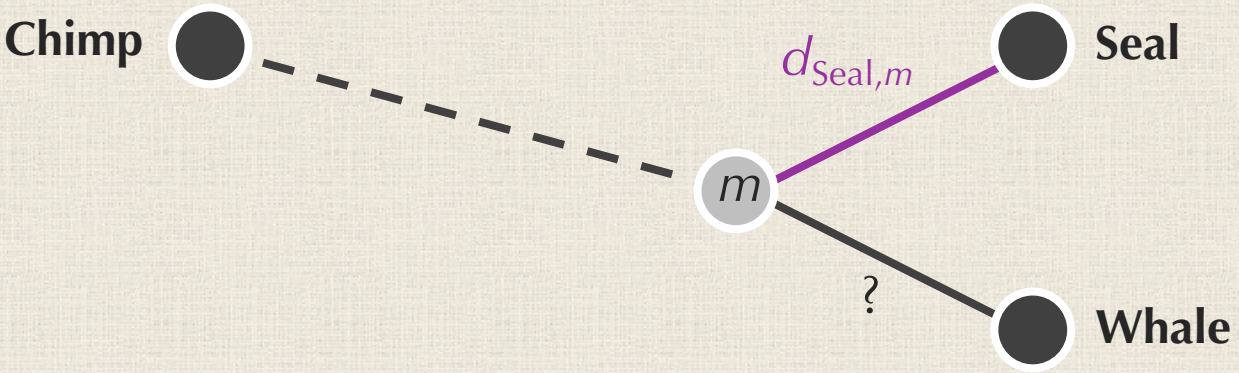
| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |




$$d_{Seal,m} = (D_{Seal,Chimp} + D_{Seal,Whale} - D_{Whale,Chimp}) / 2$$

# An Idea for Distance-Based Phylogeny

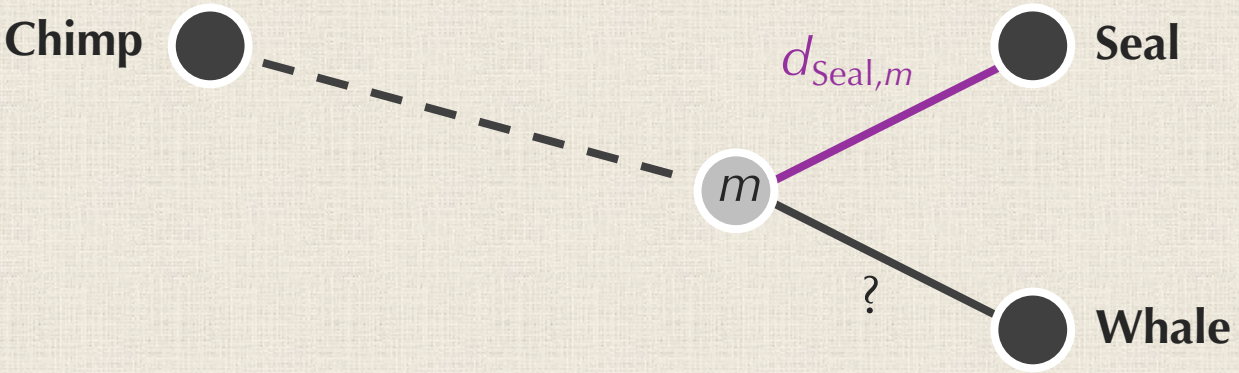
| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |




$$d_{Seal,m} = ( \quad 6 \quad + D_{Seal,Whale} - D_{Whale,Chimp}) / 2$$

# An Idea for Distance-Based Phylogeny

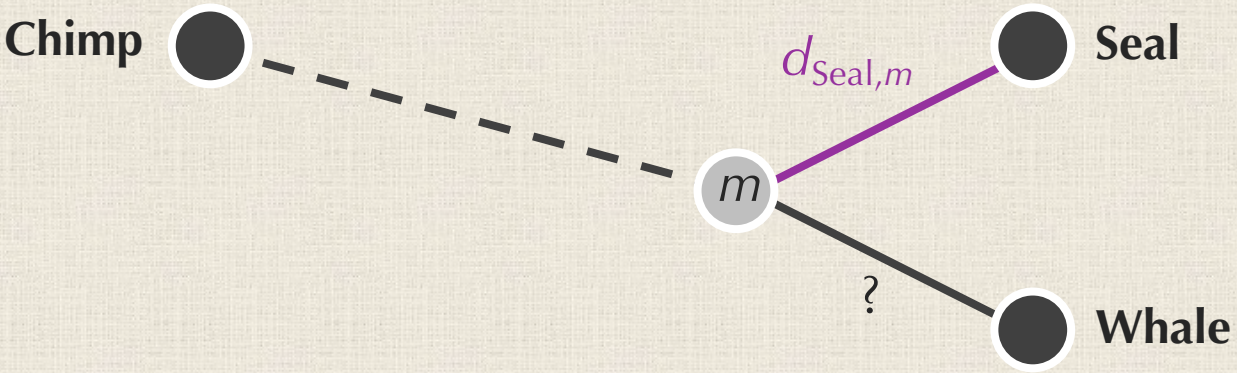
| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |




$$d_{Seal,m} = (\quad 6 \quad + \quad 2 \quad - D_{Whale,Chimp}) / 2$$

# An Idea for Distance-Based Phylogeny

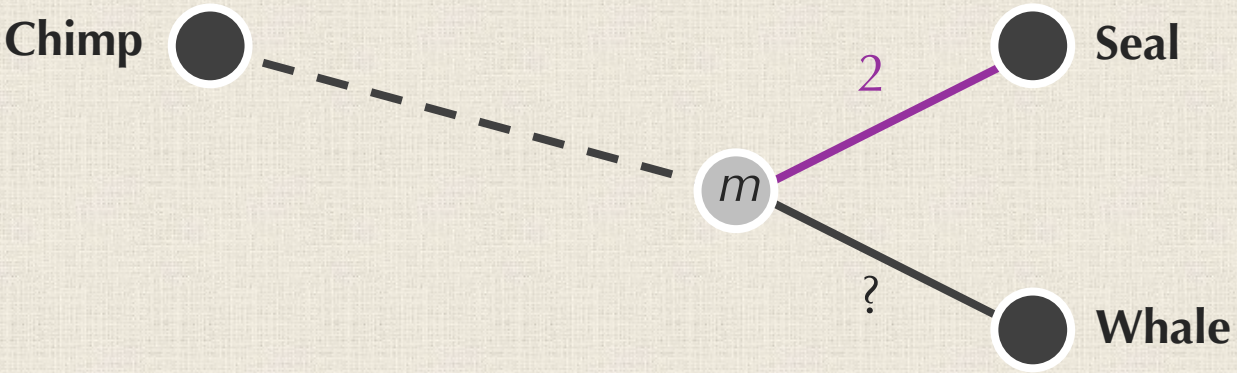
| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |




$$d_{Seal,m} = ( \quad 6 \quad + \quad 2 \quad - \quad 4 \quad ) / 2$$

# An Idea for Distance-Based Phylogeny

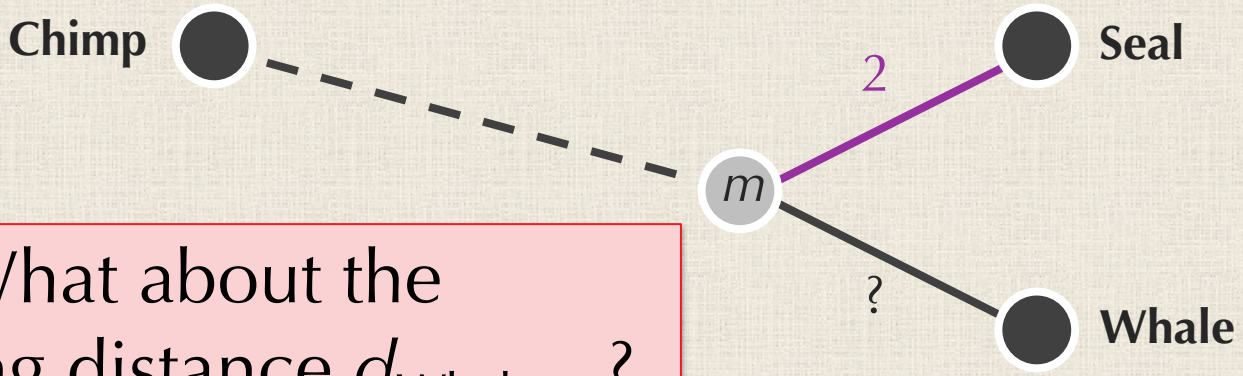
| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |




$d_{Seal,m} = 2$

# An Idea for Distance-Based Phylogeny

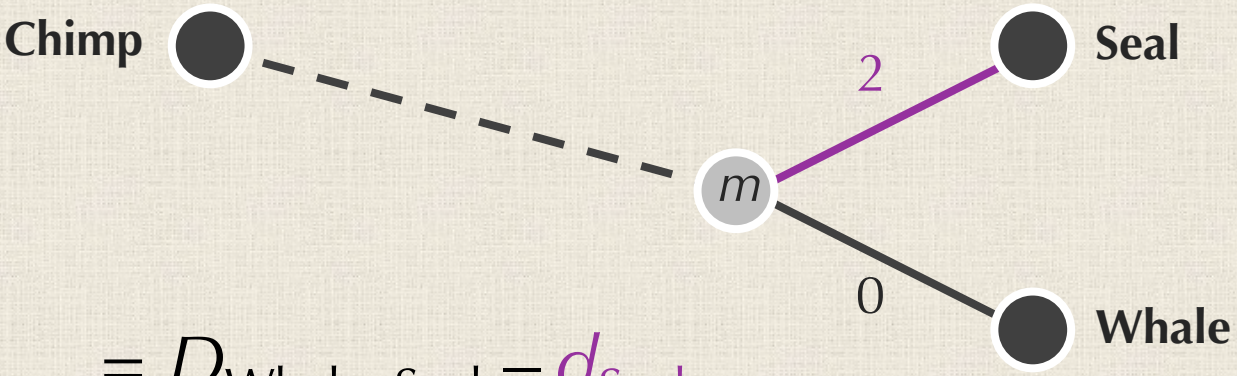
| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |




**STOP:** What about the remaining distance $d_{\text{Whale, } m}$?

# An Idea for Distance-Based Phylogeny

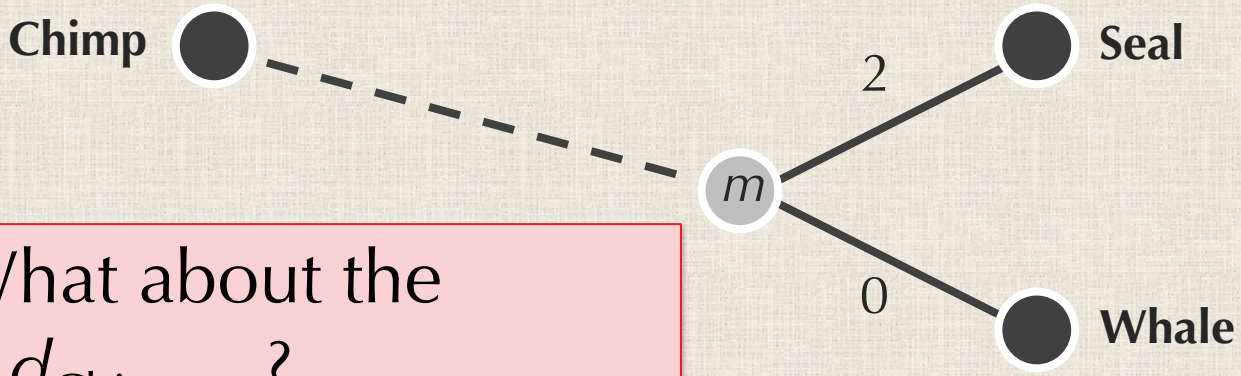
| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |




$$d_{\text{Whale, } m} = D_{\text{Whale, Seal}} - d_{\text{Seal},m}$$
$$= 2 - 2 = 0$$

# An Idea for Distance-Based Phylogeny

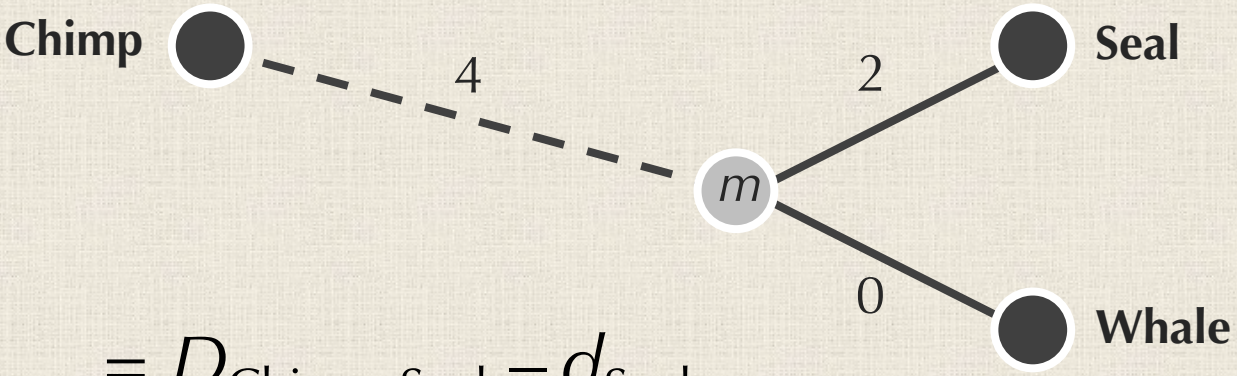
| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |




**STOP:** What about the distance $d_{\text{Chimp}, m}$?

# An Idea for Distance-Based Phylogeny

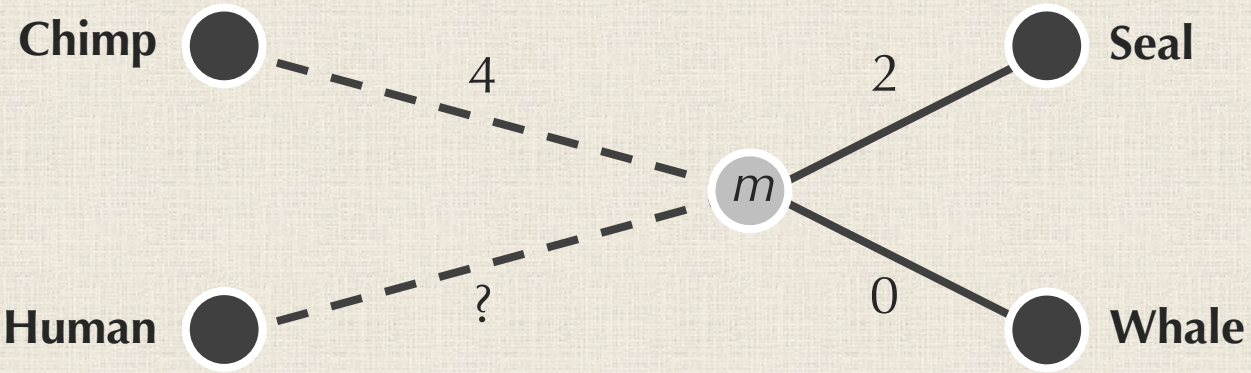
| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |




$$d_{\text{Chimp, } m} = D_{\text{Chimp, Seal}} - d_{\text{Seal},m}$$
$$= 6 - 2 = 4$$

# An Idea for Distance-Based Phylogeny

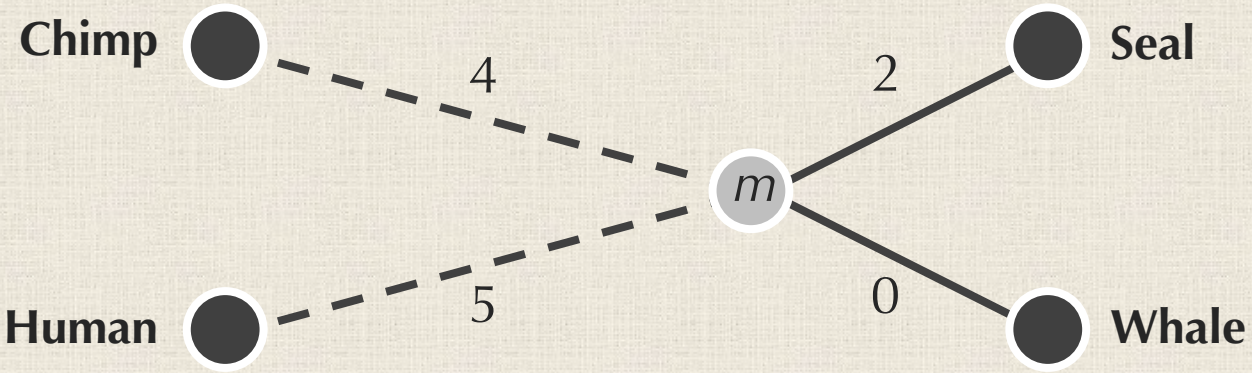
| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |




**STOP:** And $d_{\text{Human}, m}$?

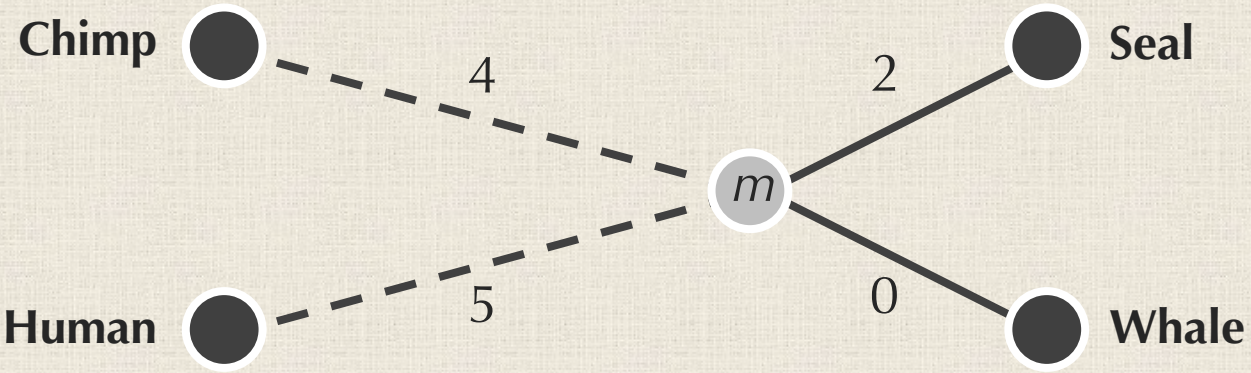
# An Idea for Distance-Based Phylogeny

| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |



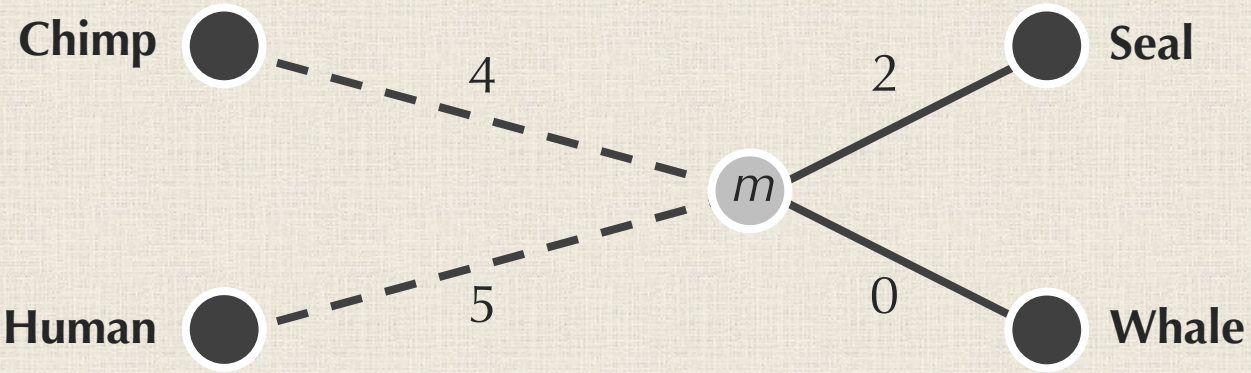
# We Add a Row and Column for *m*…

| | Chimp | Human | Seal | Whale | *m* |
|---|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 | 4 |
| **Human** | 3 | 0 | 7 | 5 | 5 |
| **Seal** | 6 | 7 | 0 | 2 | 2 |
| **Whale** | 4 | 5 | 2 | 0 | 0 |
| ***m*** | 4 | 5 | 2 | 0 | 0 |



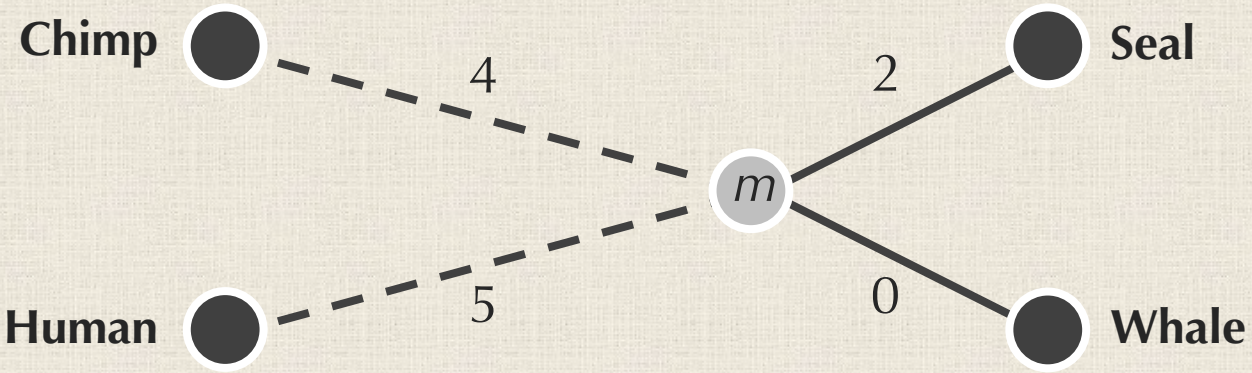
# … and recurse on this 3x3 matrix!

| | Chimp | Human | Seal | Whale | *m* |
|---|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 | 4 |
| **Human** | 3 | 0 | 7 | 5 | 5 |
| **Seal** | 6 | 7 | 0 | 2 | 2 |
| **Whale** | 4 | 5 | 2 | 0 | 0 |
| ***m*** | 4 | 5 | 2 | 0 | 0 |



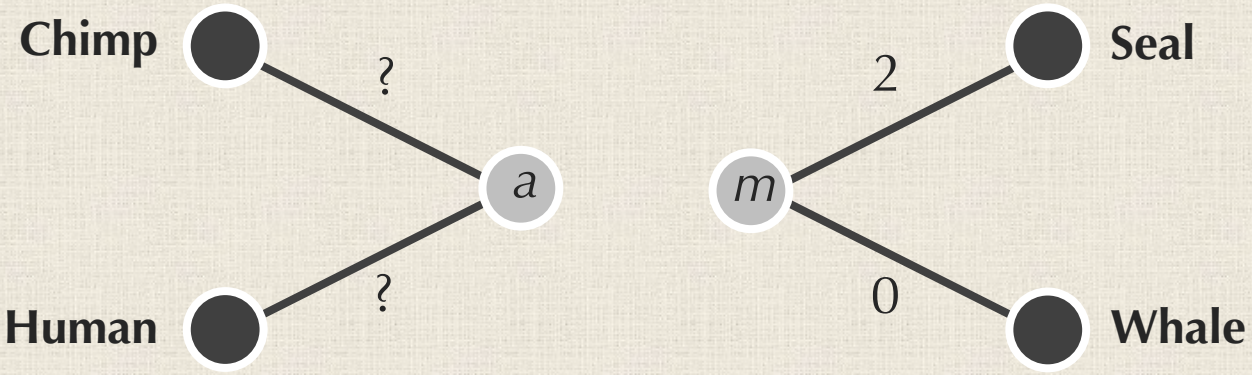
# … and recurse on this 3x3 matrix!

| | Chimp | Human | *m* |
|---|---|---|---|
| **Chimp** | 0 | 3 | 4 |
| **Human** | 3 | 0 | 5 |
| ***m*** | 4 | 5 | 0 |



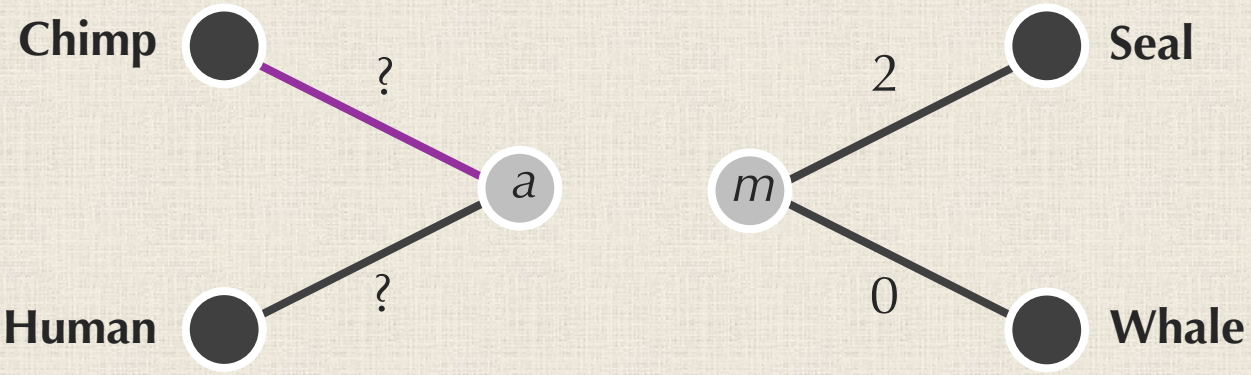
# An Idea for Distance-Based Phylogeny

| | Chimp | Human | *m* |
|---|---|---|---|
| **Chimp** | 0 | **3** | 4 |
| **Human** | **3** | 0 | 5 |
| ***m*** | 4 | 5 | 0 |



# An Idea for Distance-Based Phylogeny

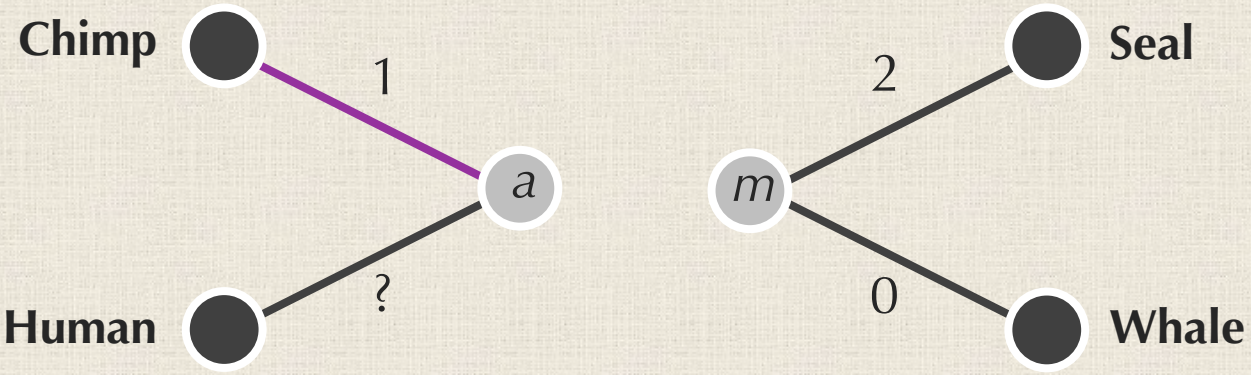
| | Chimp | Human | *m* |
|---|---|---|---|
| **Chimp** | 0 | 3 | 4 |
| **Human** | 3 | 0 | 5 |
| ***m*** | 4 | 5 | 0 |




$$d_{\text{Chimp},a} = (D_{\text{Chimp},m} + D_{\text{Chimp,Human}} - D_{\text{Human},m}) / 2$$

# An Idea for Distance-Based Phylogeny

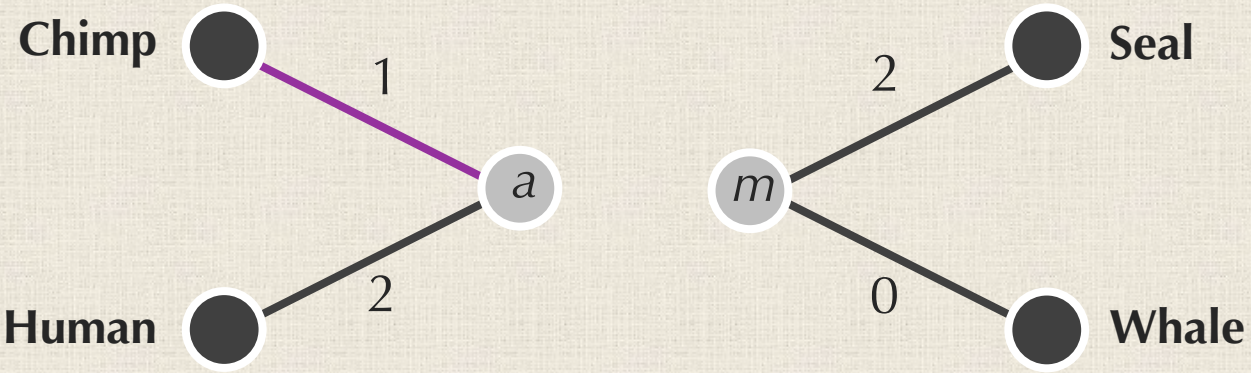
| | Chimp | Human | *m* |
|---|---|---|---|
| **Chimp** | 0 | 3 | 4 |
| **Human** | 3 | 0 | 5 |
| ***m*** | 4 | 5 | 0 |




$d_{\text{Chimp},a} = 1$

# An Idea for Distance-Based Phylogeny

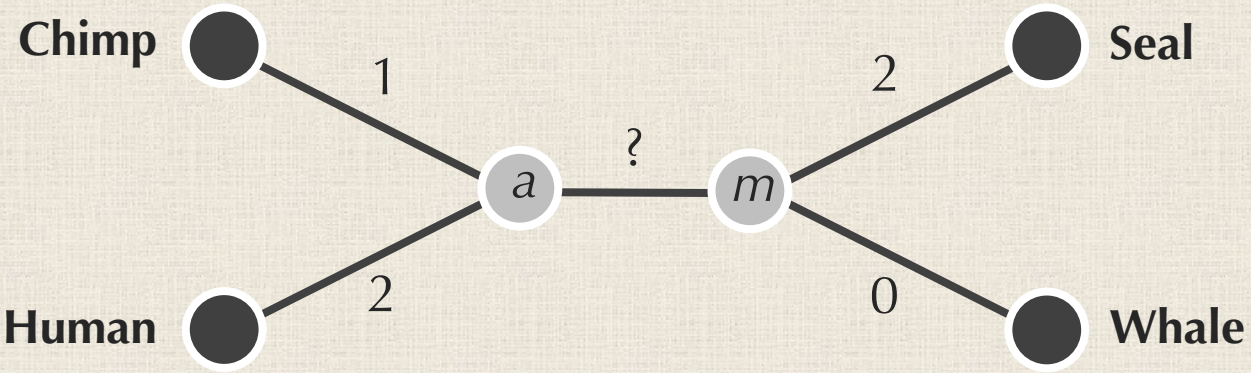
| | Chimp | Human | *m* |
|---|---|---|---|
| **Chimp** | 0 | 3 | 4 |
| **Human** | 3 | 0 | 5 |
| ***m*** | 4 | 5 | 0 |




$d_{\text{Chimp},a} = 1$, so $d_{\text{Human},a} = d_{\text{Human, Chimp}} - d_{\text{Chimp},a} = 2$

# An Idea for Distance-Based Phylogeny

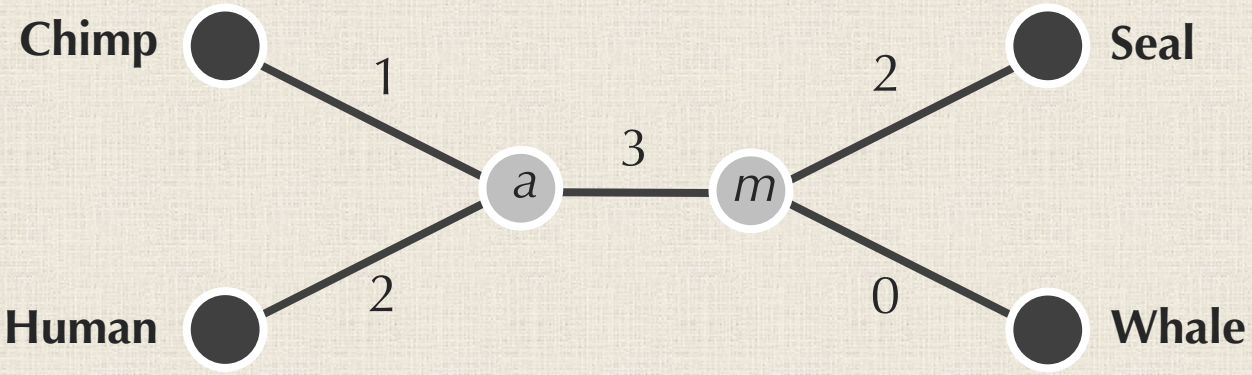
| | Chimp | Human | *m* |
|---|---|---|---|
| **Chimp** | 0 | 3 | 4 |
| **Human** | 3 | 0 | 5 |
| ***m*** | 4 | 5 | 0 |




**STOP:** And $d_{a,\,m}$?

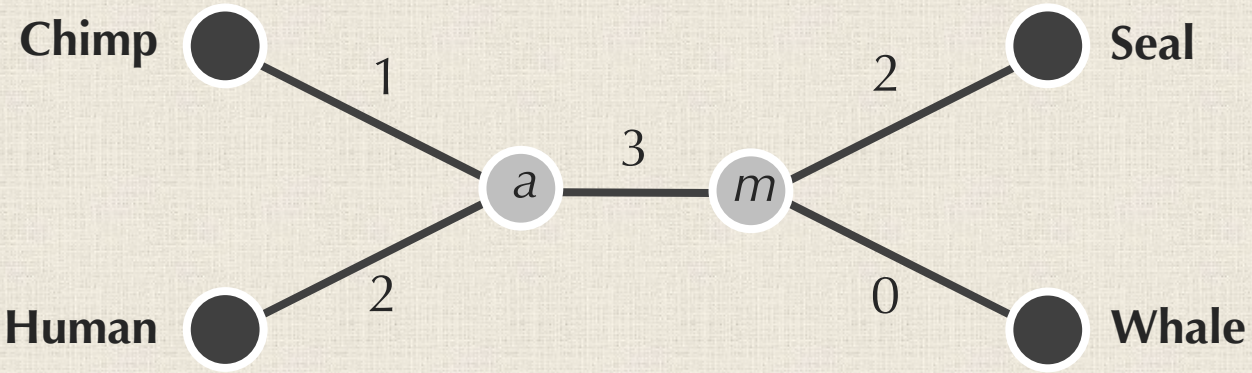
# An Idea for Distance-Based Phylogeny

| | Chimp | Human | *m* |
|---|---|---|---|
| **Chimp** | 0 | 3 | 4 |
| **Human** | 3 | 0 | 5 |
| ***m*** | 4 | 5 | 0 |



# Note that the Matrix Fits the Tree

| | Chimp | Human | Seal | Whale |
|---|---|---|---|---|
| **Chimp** | 0 | 3 | 6 | 4 |
| **Human** | 3 | 0 | 7 | 5 |
| **Seal** | 6 | 7 | 0 | 2 |
| **Whale** | 4 | 5 | 2 | 0 |



# ULTRAMETRIC TREES AND UPGMA

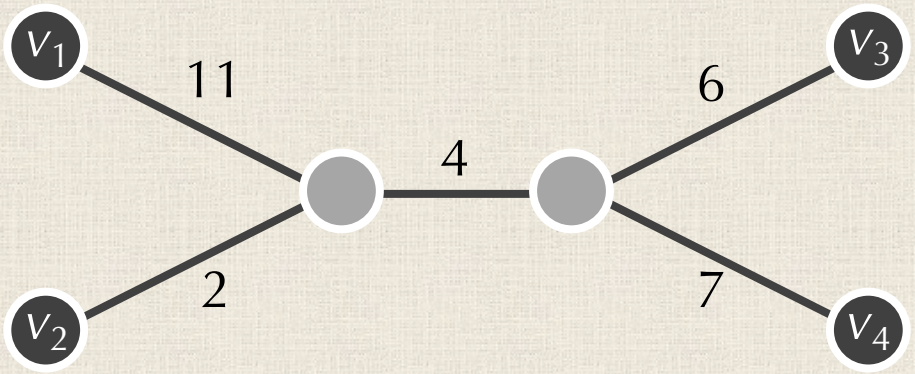
# An Idea for Distance-Based Phylogeny

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |

**Exercise:** Apply our recursive approach to this additive distance matrix.

# What Was Wrong With Our Algorithm?

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |

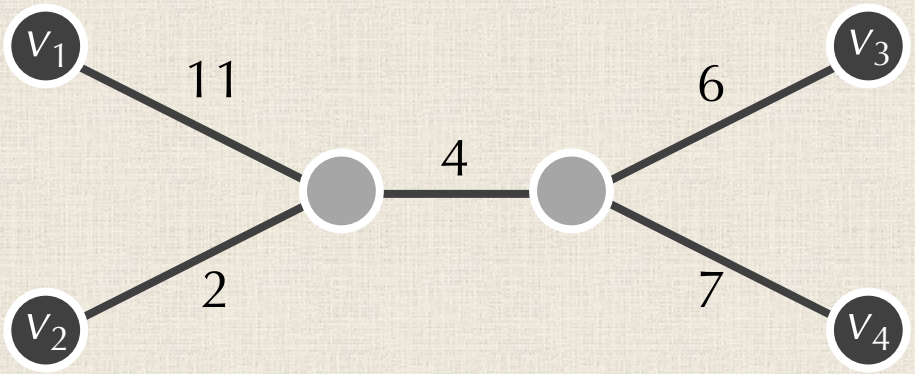



*Here is the tree fitting the matrix...*

# What Was Wrong With Our Algorithm?

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | **12** | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |

minimum element is $D_{2,3}$

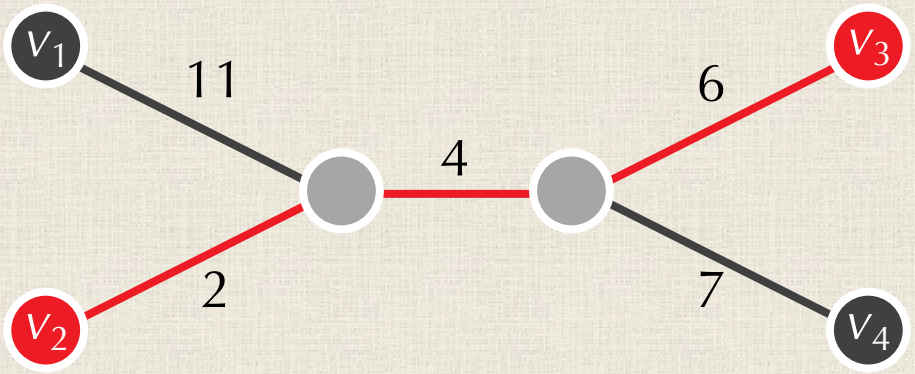



*Here is the tree fitting the matrix…*

# What Was Wrong With Our Algorithm?

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | **12** | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |

minimum element is $D_{2,3}$

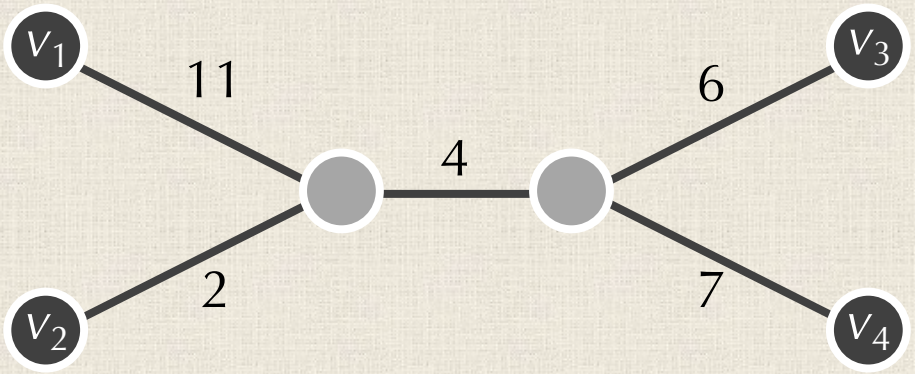



$v_2$ and $v_3$ are **not** neighbors!

# What Was Wrong With Our Algorithm?

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |

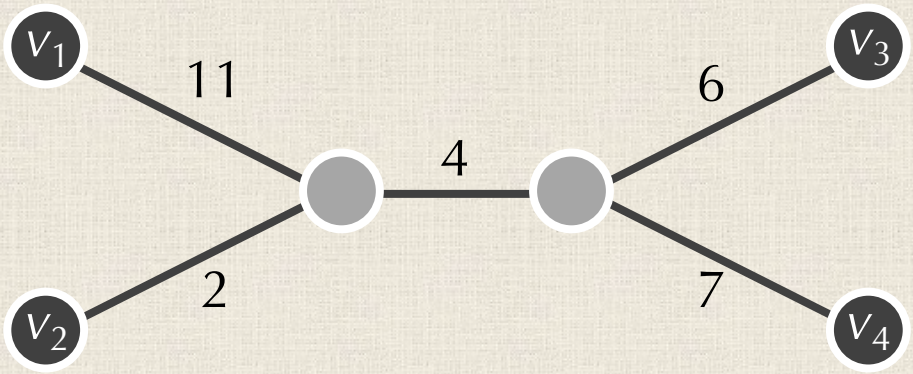
**STOP:** What about the tree prevents the minimum matrix element from corresponding to neighbors?



# What Was Wrong With Our Algorithm?

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |

**STOP:** What about the tree prevents the minimum matrix element from corresponding to neighbors?




**Answer:** $v_1$ is an *outlier*, so the distance from $v_1$ to any other node is higher than average.

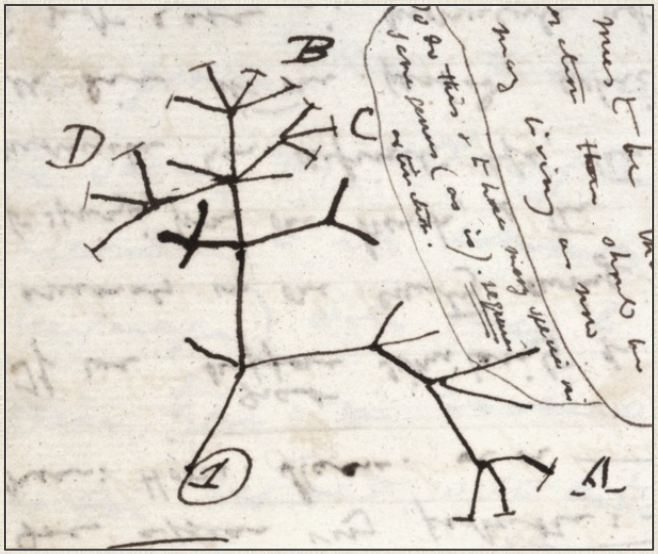
# Modeling Speciations

Even though the minimum element of $D$ doesn't necessarily correspond to neighbors, this assumption powers one of the most famous evolutionary tree heuristics.

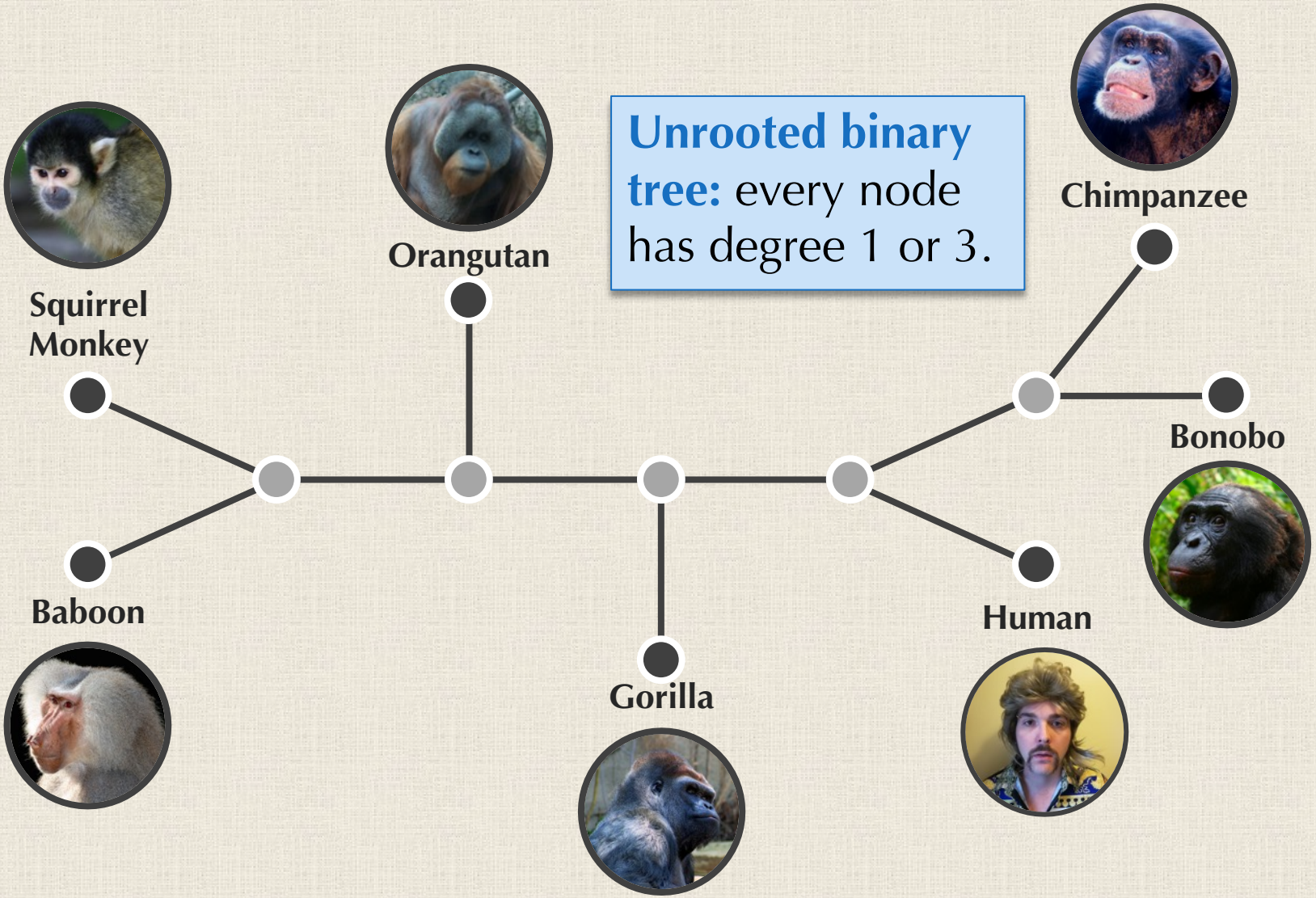
# Modeling Speciations

Even though the minimum element of *D* doesn't necessarily correspond to neighbors, this assumption powers one of the most famous evolutionary tree heuristics.

Researchers often assume that all internal nodes correspond to **speciations**, where one species splits into two.

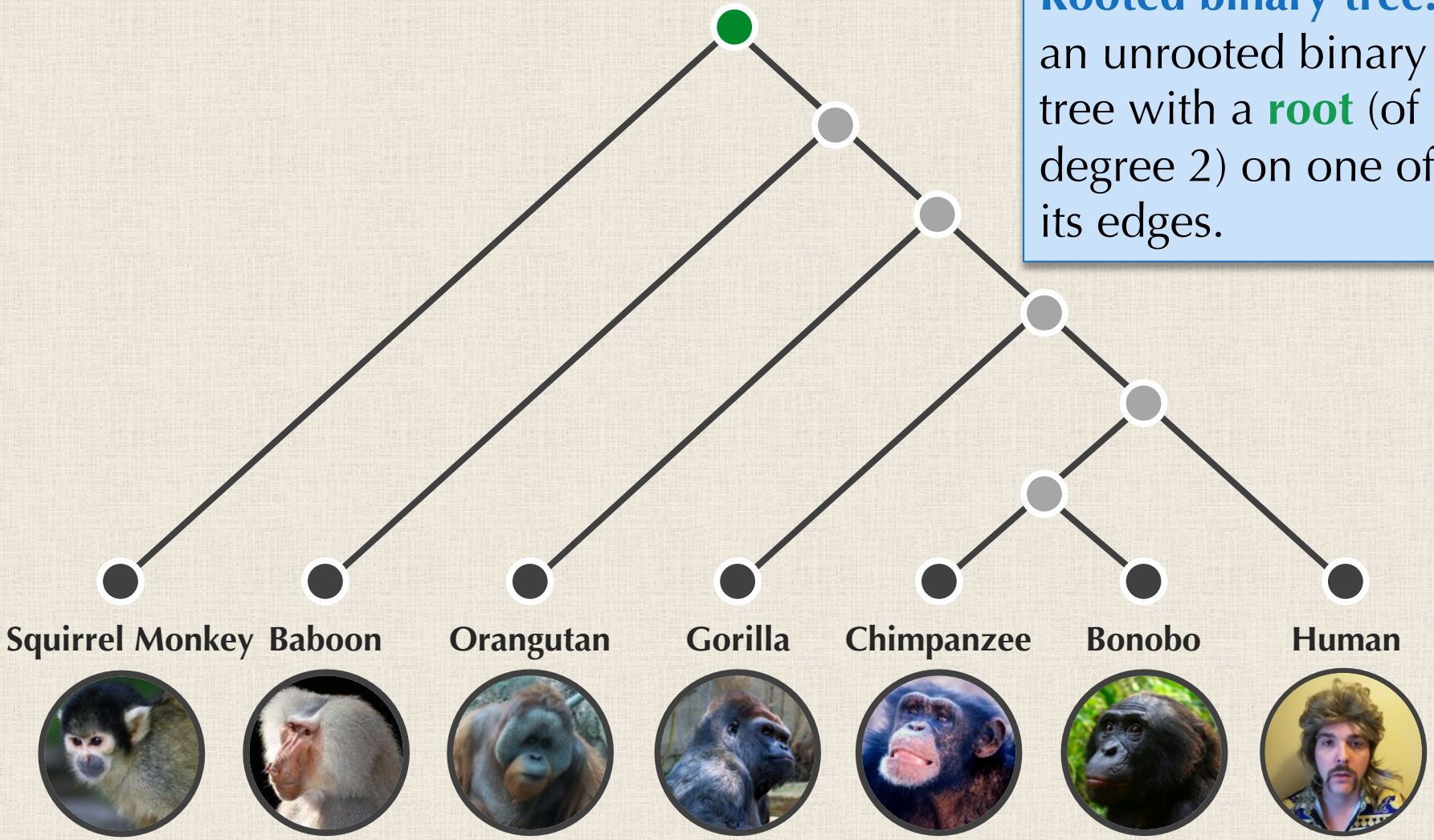


# We've Thus Far Seen *Unrooted* Trees



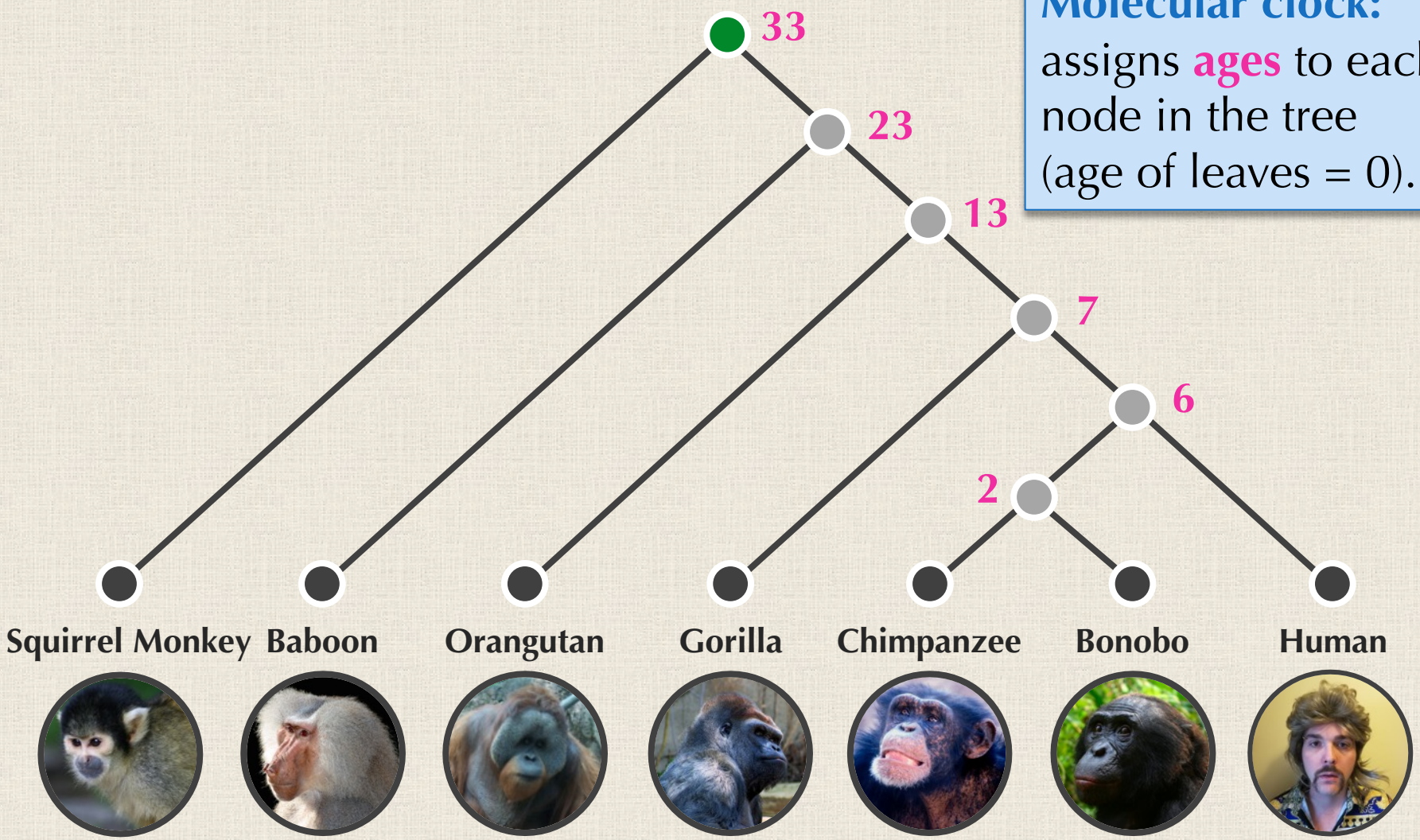
# From *Unrooted* to *Rooted* Trees

**Rooted binary tree:** an unrooted binary tree with a **root** (of degree 2) on one of its edges.

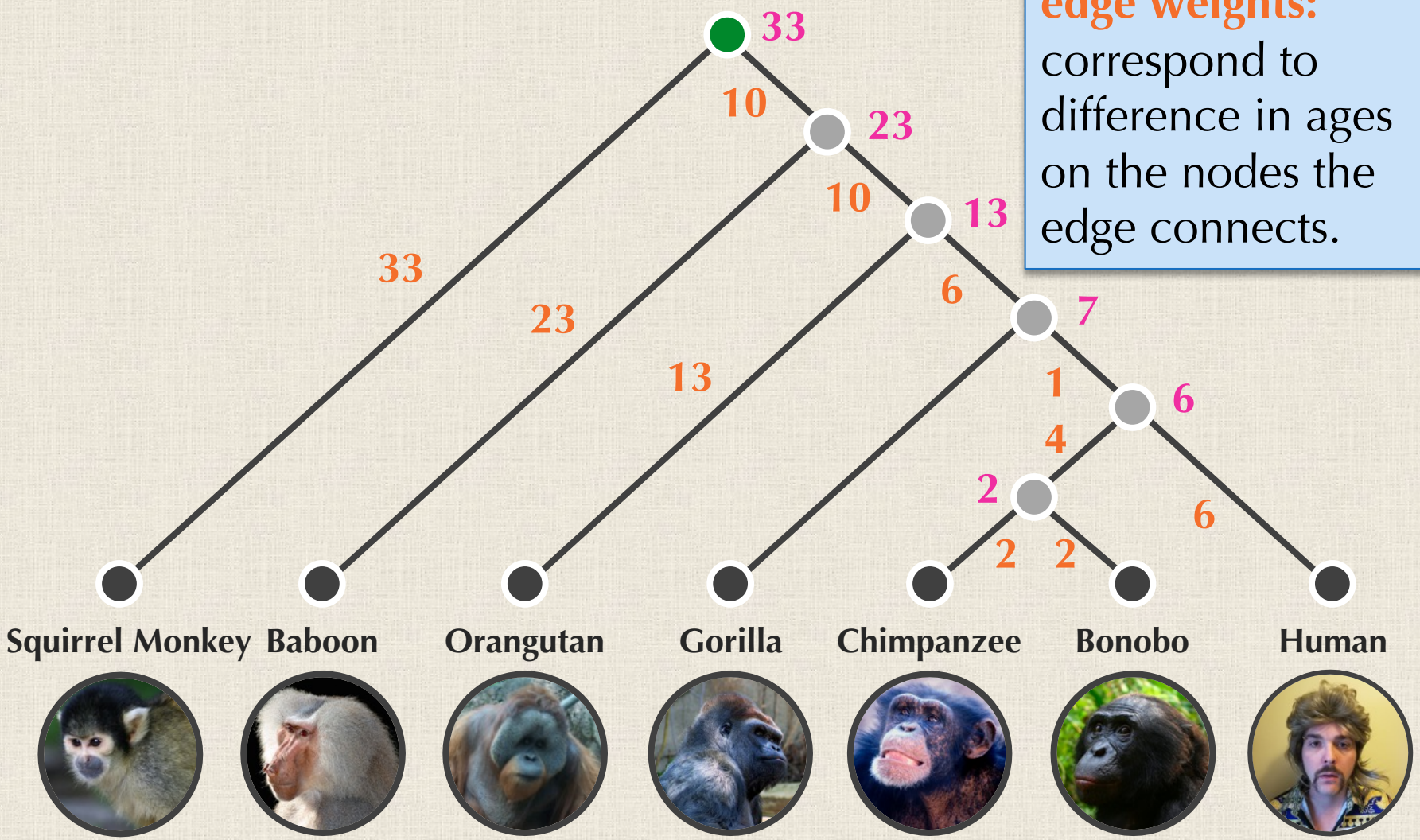


# From *Unrooted* to *Rooted* Trees

**Molecular clock:**
assigns **ages** to each node in the tree (age of leaves = 0).



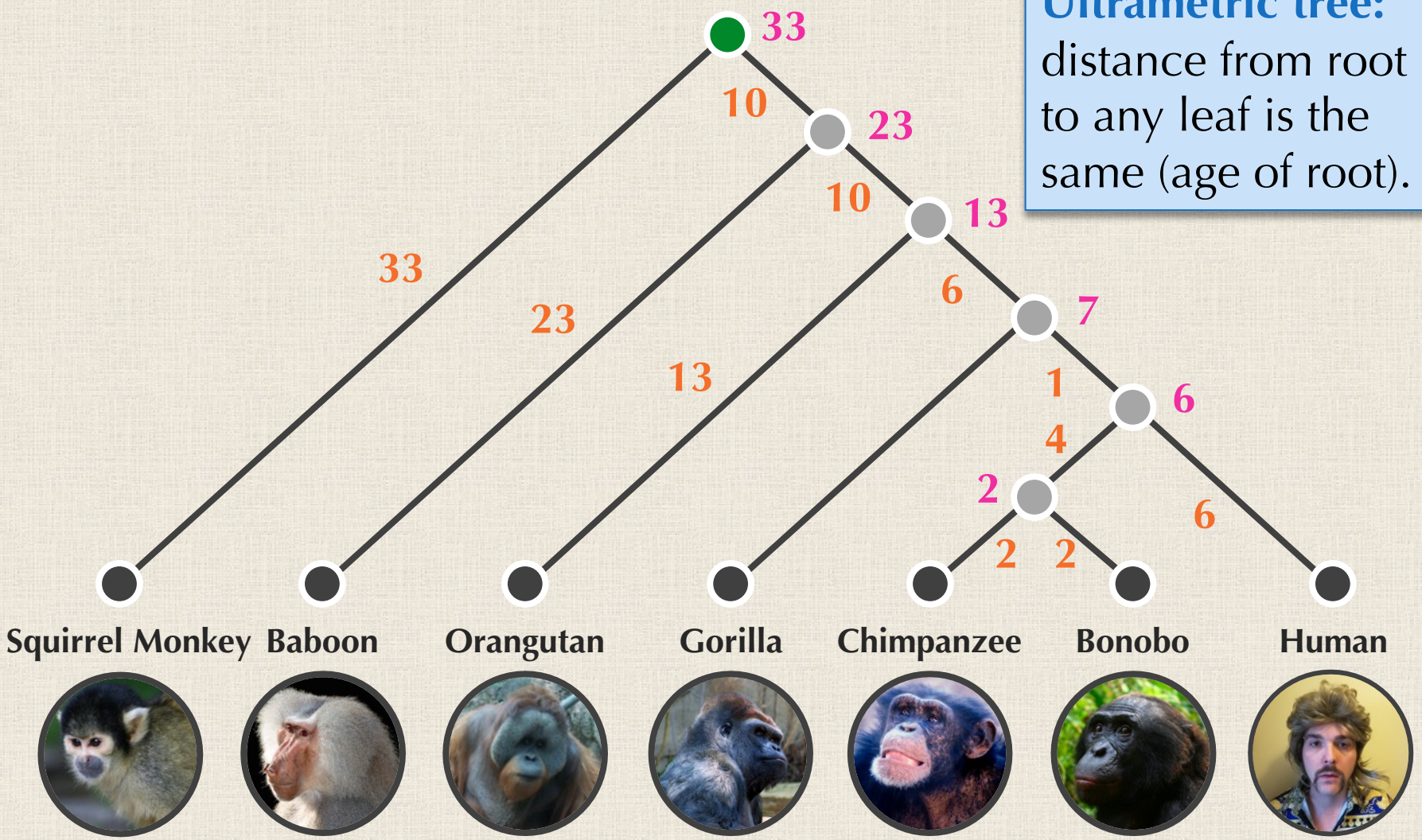
# From *Unrooted* to *Rooted* Trees




**edge weights:** correspond to difference in ages on the nodes the edge connects.

# From *Unrooted* to *Rooted* Trees

**Ultrametric tree:** distance from root to any leaf is the same (age of root).



# UPGMA: A Clustering Heuristic

1. Form a cluster for each present-day species, each containing a single leaf.

| | *i* | *j* | *k* | *l* |
|---|---|---|---|---|
| ***i*** | 0 | 3 | 4 | 3 |
| ***j*** | 3 | 0 | 4 | 5 |
| ***k*** | 4 | 4 | 0 | 2 |
| ***l*** | 3 | 5 | 2 | 0 |

# UPGMA: A Clustering Heuristic

2. Find the two closest clusters $C_1$ and $C_2$ according to the minimum value in the current matrix.

| | *i* | *j* | *k* | *l* |
|---|---|---|---|---|
| ***i*** | 0 | 3 | 4 | 3 |
| ***j*** | 3 | 0 | 4 | 5 |
| ***k*** | 4 | 4 | 0 | **2** |
| ***l*** | 3 | 5 | **2** | 0 |

# UPGMA: A Clustering Heuristic

3. Merge $C_1$ and $C_2$ into a single cluster $C$.

| | *i* | *j* | *k* | *l* |
|---|---|---|---|---|
| ***i*** | 0 | 3 | 4 | 3 |
| ***j*** | 3 | 0 | 4 | 5 |
| ***k*** | 4 | 4 | 0 | **2** |
| ***l*** | 3 | 5 | **2** | 0 |

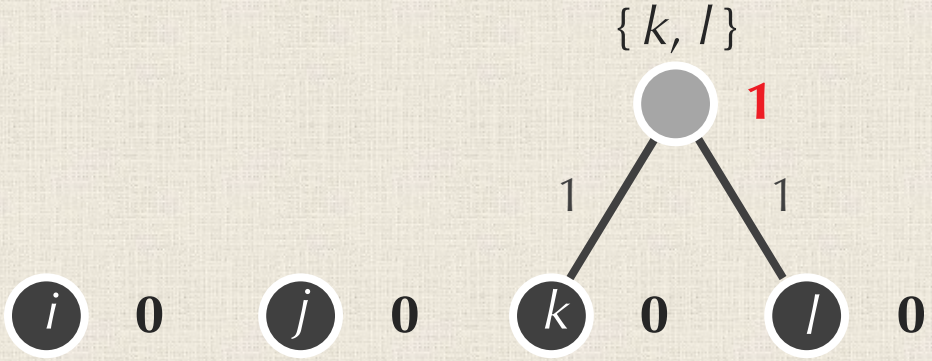
$\{k, l\}$

*i* **0** *j* **0** *k* **0** *l* **0**

# UPGMA: A Clustering Heuristic

4. Form a new node for $C$ and connect to $C_1$ and $C_2$ by an edge. Set age of $C$ as $D(C_1, C_2)/2$.

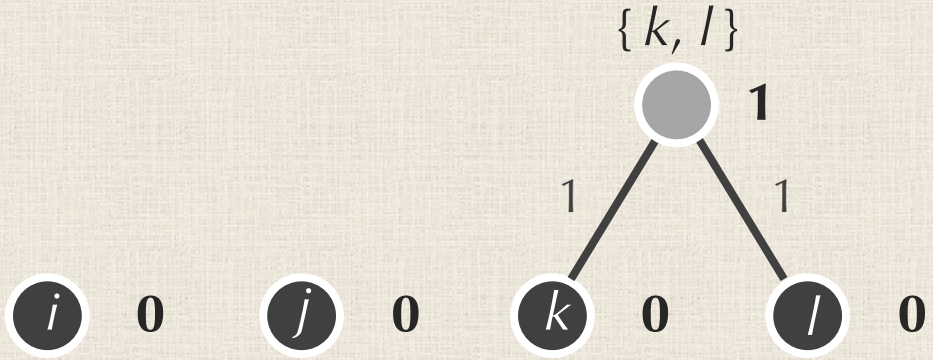
| | *i* | *j* | *k* | *l* |
|---|---|---|---|---|
| ***i*** | 0 | 3 | 4 | 3 |
| ***j*** | 3 | 0 | 4 | 5 |
| ***k*** | 4 | 4 | 0 | **2** |
| ***l*** | 3 | 5 | **2** | 0 |



# UPGMA: A Clustering Heuristic

5. Update the distance matrix by computing the average distance between each pair of clusters.

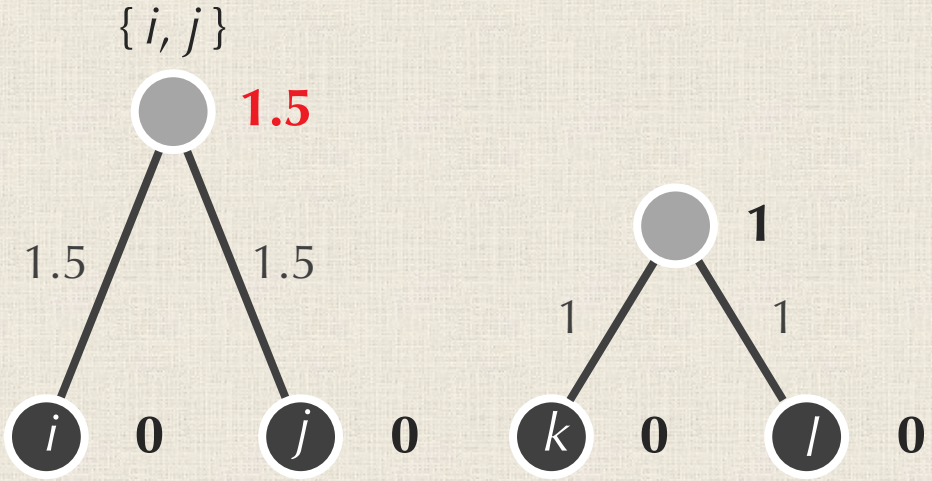
| | ***i*** | ***j*** | **{*k*, *l*}** |
|---|---|---|---|
| ***i*** | 0 | 3 | 3.5 |
| ***j*** | 3 | 0 | 4.5 |
| **{*k*, *l*}** | 3.5 | 4.5 | 0 |



# UPGMA: A Clustering Heuristic

6. Iterate steps 2-5 until a single cluster contains all species.

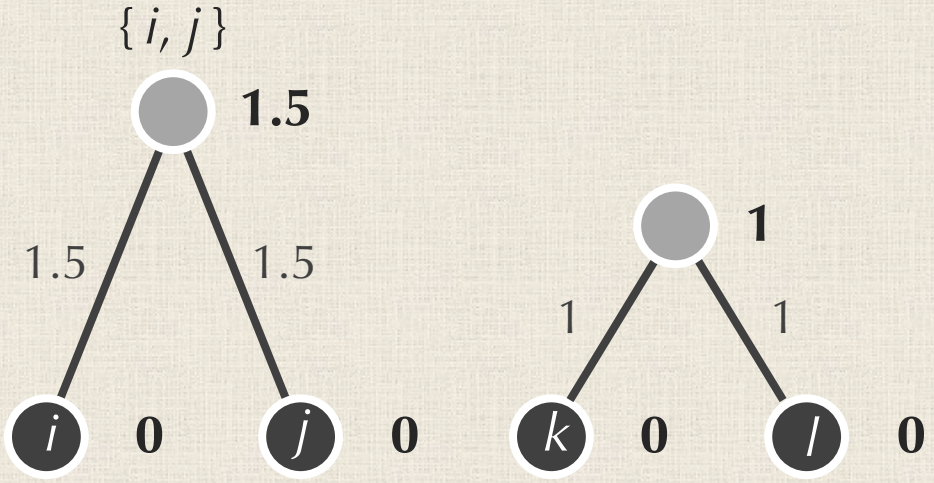
| | $i$ | $j$ | $\{k, l\}$ |
|---|---|---|---|
| $i$ | 0 | **3** | 3.5 |
| $j$ | **3** | 0 | 4.5 |
| $\{k, l\}$ | 3.5 | 4.5 | 0 |



# UPGMA: A Clustering Heuristic

6. Iterate steps 2-5 until a single cluster contains all species.

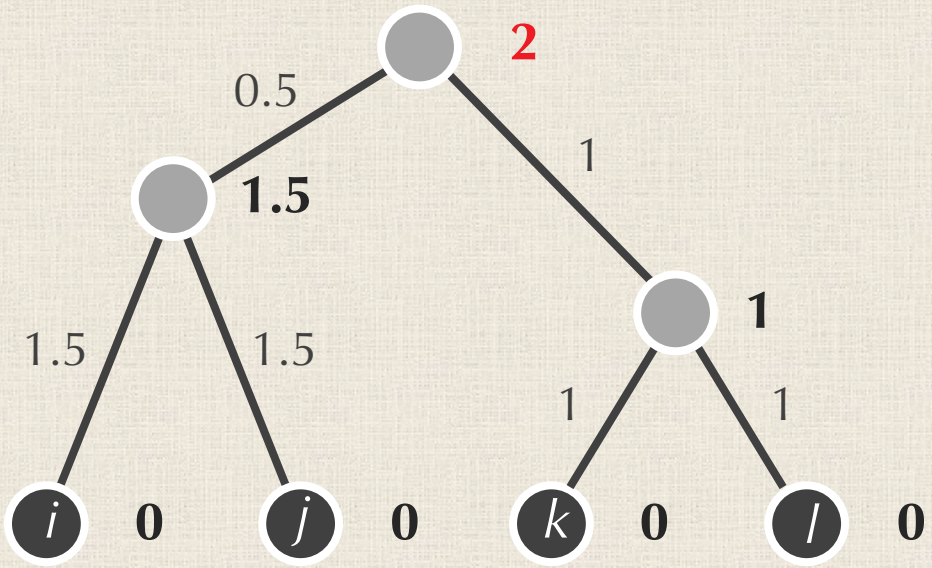
| | $\{i, j\}$ | $\{k, l\}$ |
|---|---|---|
| $\{i, j\}$ | 0 | 4 |
| $\{k, l\}$ | 4 | 0 |



# UPGMA: A Clustering Heuristic

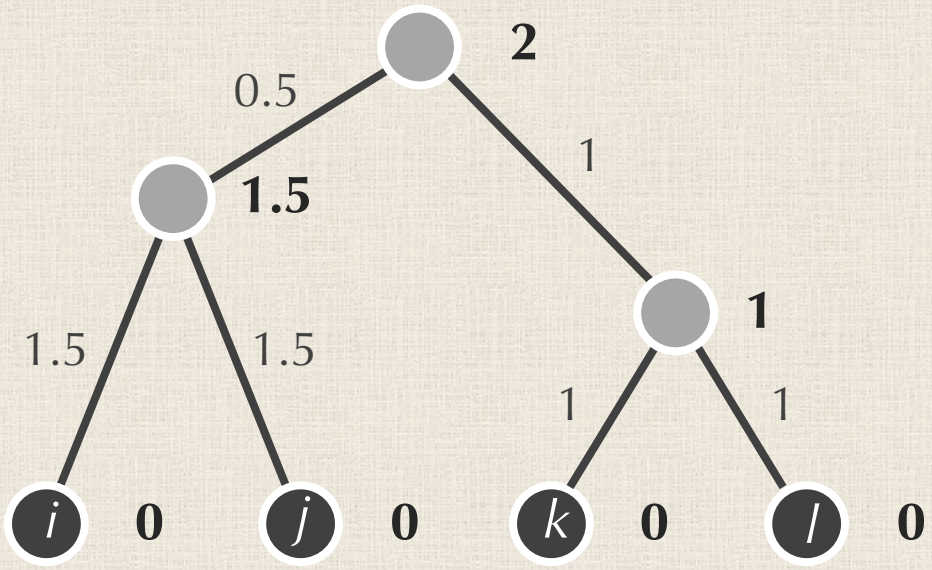
6. Iterate steps 2-5 until a single cluster contains all species.

| | $\{i, j\}$ | $\{k, l\}$ |
|---|---|---|
| $\{i, j\}$ | 0 | **4** |
| $\{k, l\}$ | **4** | 0 |



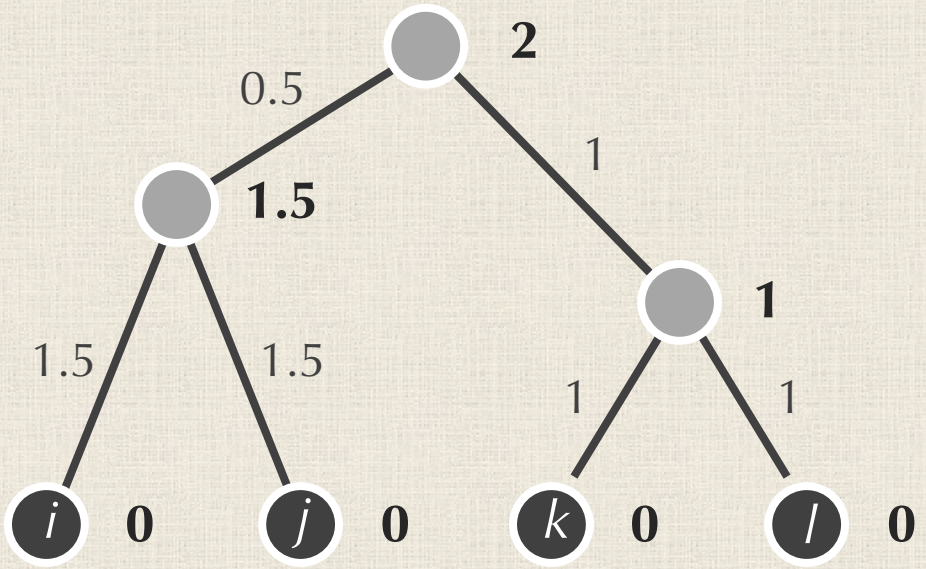
# UPGMA: A Clustering Heuristic

6. Iterate steps 2-5 until a single cluster contains all species.



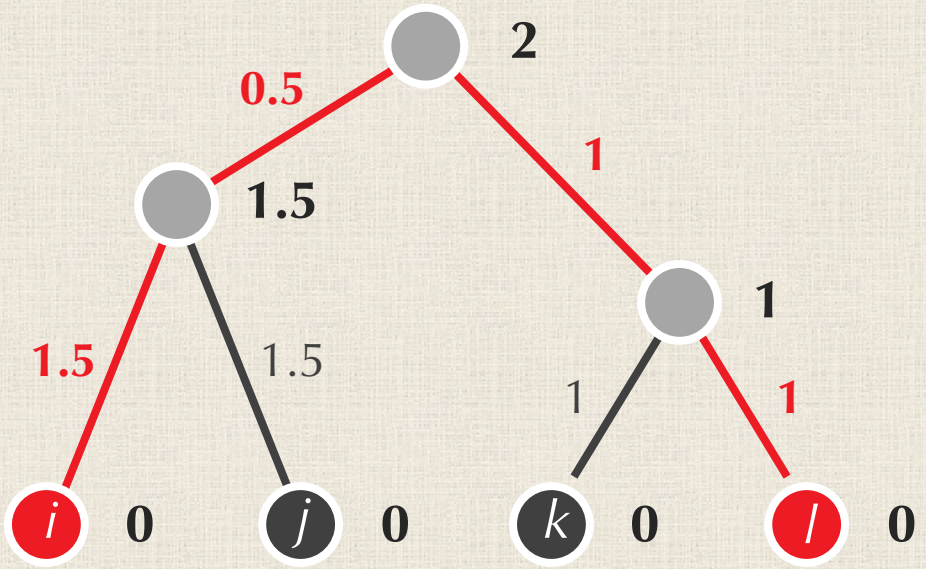
# UPGMA Doesn't "Fit" a Tree to a Matrix

| | *i* | *j* | *k* | *l* |
|---|---|---|---|---|
| ***i*** | 0 | 3 | 4 | 3 |
| ***j*** | 3 | 0 | 4 | 5 |
| ***k*** | 4 | 4 | 0 | 2 |
| ***l*** | 3 | 5 | 2 | 0 |



# UPGMA Doesn't "Fit" a Tree to a Matrix

| | *i* | *j* | *k* | *l* |
|---|---|---|---|---|
| ***i*** | 0 | 3 | 4 | **3** |
| ***j*** | 3 | 0 | 4 | 5 |
| ***k*** | 4 | 4 | 0 | 2 |
| ***l*** | 3 | 5 | 2 | 0 |



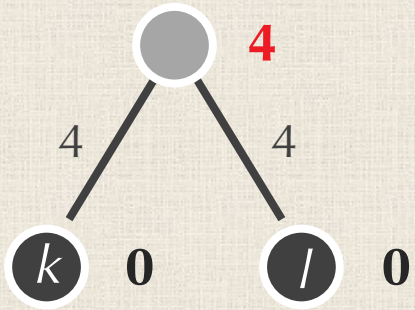
# Quick UPGMA Quiz

**Exercise:** Apply UPGMA to the following matrix.

| | *i* | *j* | *k* | *l* |
|---|---|---|---|---|
| ***i*** | 0 | 20 | 9 | 11 |
| ***j*** | 20 | 0 | 17 | 11 |
| ***k*** | 9 | 17 | 0 | 8 |
| ***l*** | 11 | 11 | 8 | 0 |

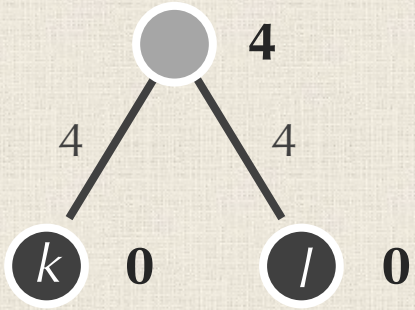
# Quick UPGMA Quiz

| | *i* | *j* | *k* | *l* |
|---|---|---|---|---|
| ***i*** | 0 | 20 | 9 | 11 |
| ***j*** | 20 | 0 | 17 | 11 |
| ***k*** | 9 | 17 | 0 | **8** |
| ***l*** | 11 | 11 | **8** | 0 |



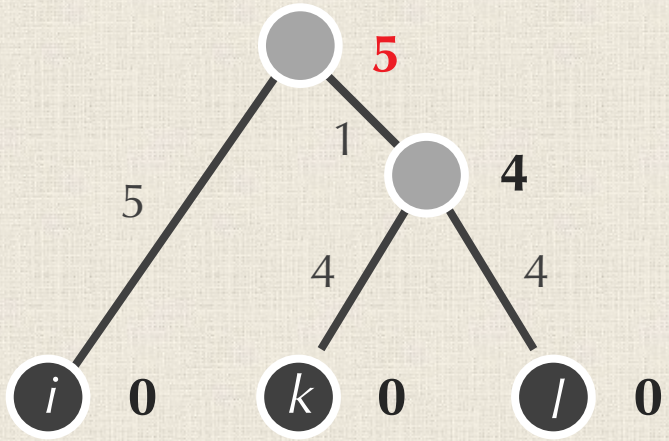
# Quick UPGMA Quiz

| | $i$ | $j$ | $\{k, l\}$ |
|---|---|---|---|
| $i$ | 0 | 20 | 10 |
| $j$ | 20 | 0 | 14 |
| $\{k, l\}$ | 10 | 14 | 0 |



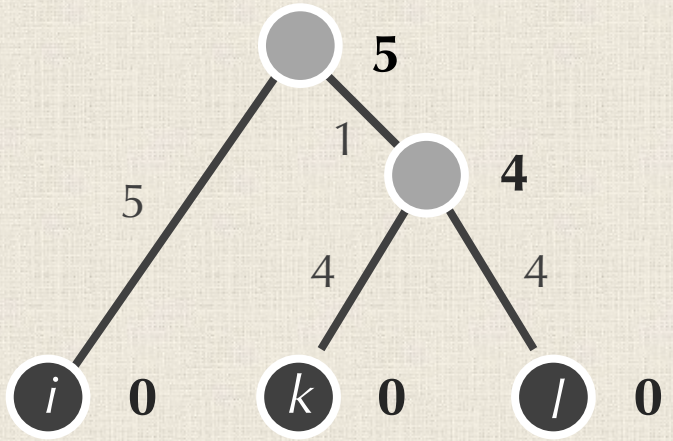
# Quick UPGMA Quiz

| | *i* | *j* | {*k*, *l*} |
|---|---|---|---|
| *i* | 0 | 20 | **10** |
| *j* | 20 | 0 | 14 |
| {*k*, *l*} | **10** | 14 | 0 |



# Quick UPGMA Quiz

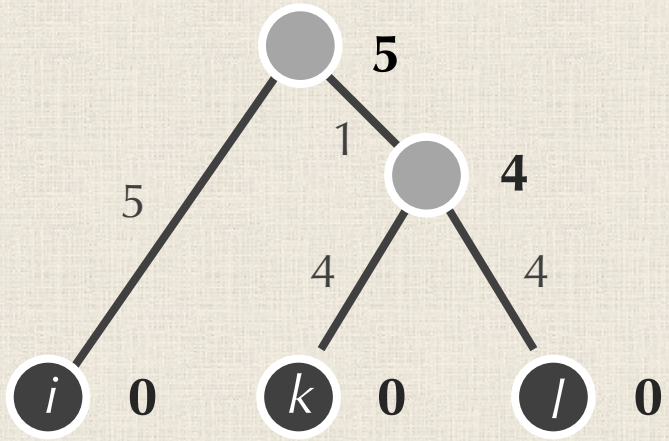
| | $j$ | $\{i, k, l\}$ |
|---|---|---|
| $j$ | 0 | 17 |
| $\{i, k, l\}$ | 17 | 0 |



# Quick UPGMA Quiz

**STOP:** This is wrong. Why?

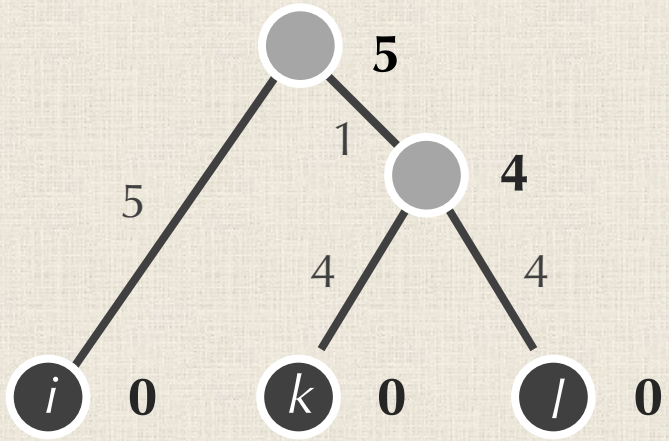
| | $j$ | $\{i, k, l\}$ |
|---|---|---|
| $j$ | 0 | 17 |
| $\{i, k, l\}$ | 17 | 0 |



# Average Distance Must be *Weighted*

**Answer:** The average distance from *j* to *i*, *k*, and *l* is (20+17+11)/3 = 16, not 17.

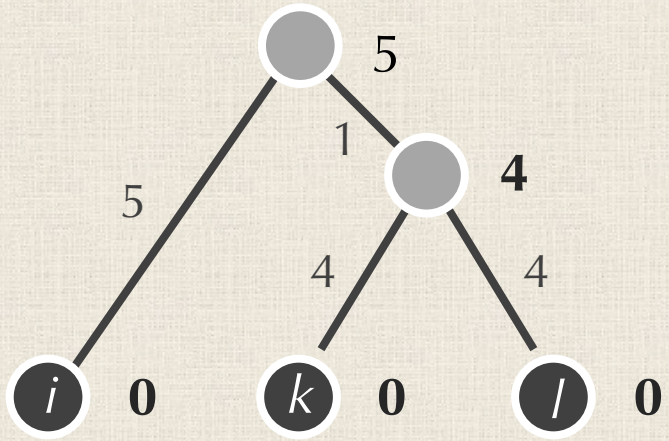
| | *i* | *j* | *k* | *l* |
|---|---|---|---|---|
| ***i*** | 0 | **20** | 9 | 11 |
| ***j*** | 20 | 0 | 17 | 11 |
| ***k*** | 9 | **17** | 0 | 8 |
| ***l*** | 11 | **11** | 8 | 0 |



# Average Distance Must be *Weighted*

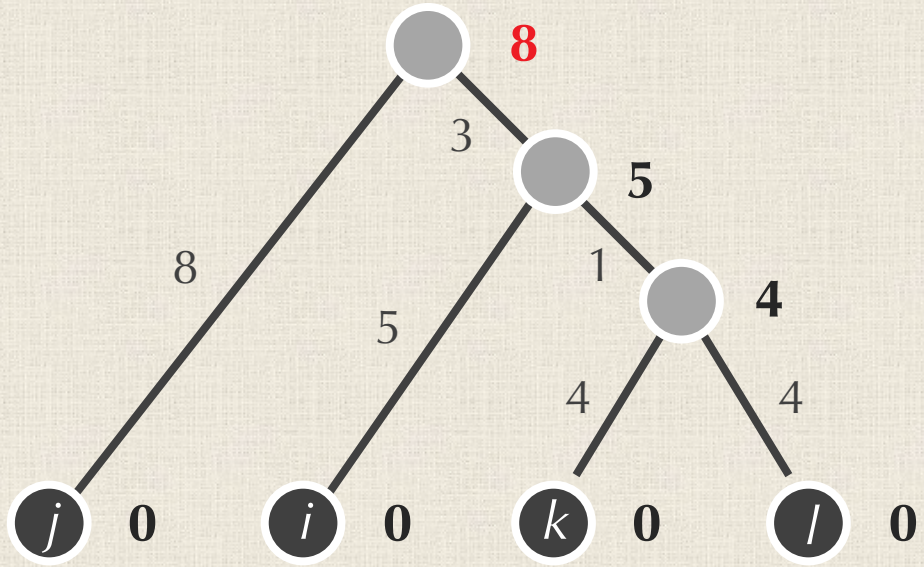
**Answer:** The average distance from *j* to *i*, *k*, and *l* is (20+17+11)/3 = 16, not 17.

| | *i* | *j* | {*k*, *l*} |
|---|---|---|---|
| ***i*** | 0 | **20** | 10 |
| ***j*** | 20 | 0 | 14 |
| **{*k*, *l*}** | 10 | **14** | 0 |



# Average Distance Must be *Weighted*

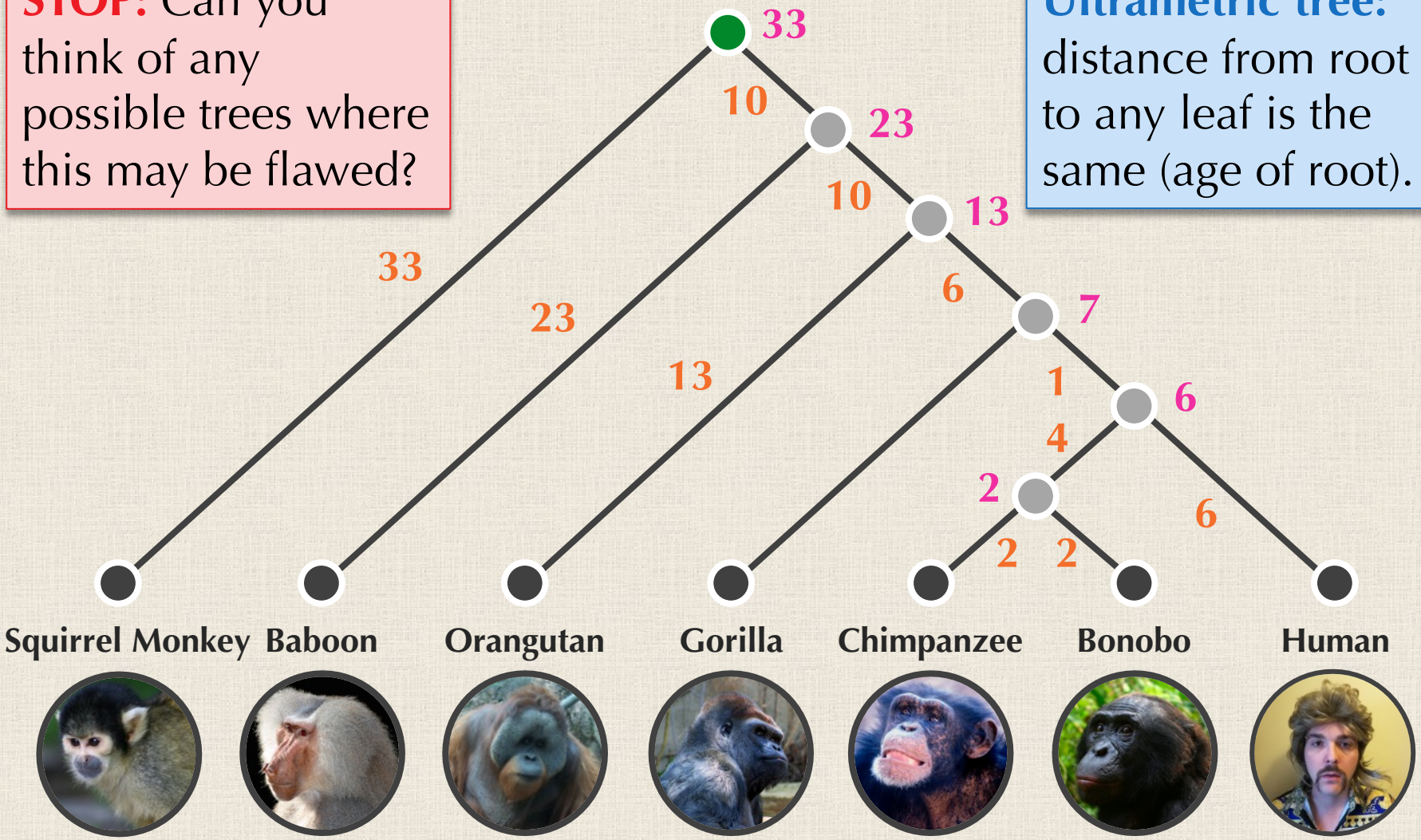
| | $j$ | $\{i, k, l\}$ |
|---|---|---|
| $j$ | 0 | **16** |
| $\{i, k, l\}$ | **16** | 0 |



# The Ultrametric Assumption is Very Strong

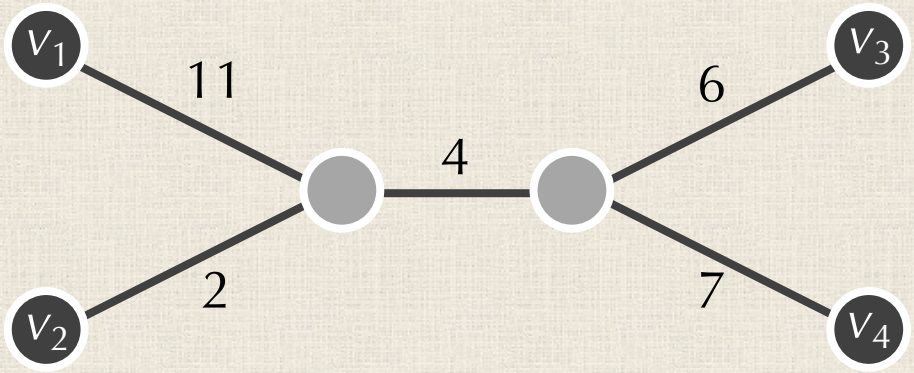
**STOP:** Can you think of any possible trees where this may be flawed?

**Ultrametric tree:** distance from root to any leaf is the same (age of root).



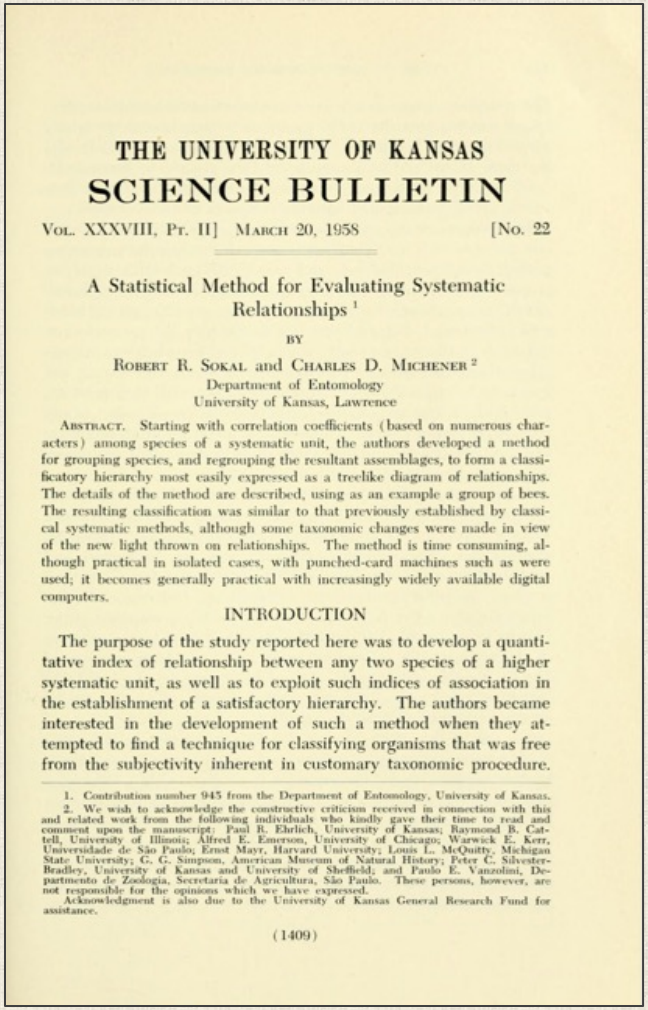
# UPGMA May Even Fail on an Additive Matrix

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |




Furthermore, this is an additive matrix, and yet UPGMA will join $v_2$ and $v_3$, producing the *wrong* tree even though there is one that perfectly fits it!

# UPGMA is Suboptimal But Has Become "Machine Learning"…



THE UNIVERSITY OF KANSAS
SCIENCE BULLETIN

VOL. XXXVIII, PT. II] MARCH 20, 1958 [NO. 22

A Statistical Method for Evaluating Systematic Relationships [1]

BY

ROBERT R. SOKAL and CHARLES D. MICHENER [2]
Department of Entomology
University of Kansas, Lawrence

ABSTRACT. Starting with correlation coefficients (based on numerous characters) among species of a systematic unit, the authors developed a method for grouping species, and regrouping the resultant assemblages, to form a classificatory hierarchy most easily expressed as a treelike diagram of relationships. The details of the method are described, using as an example a group of bees. The resulting classification was similar to that previously established by classical systematic methods, although some taxonomic changes were made in view of the new light thrown on relationships. The method is time consuming, although practical in isolated cases, with punched-card machines such as were used; it becomes generally practical with increasingly widely available digital computers.

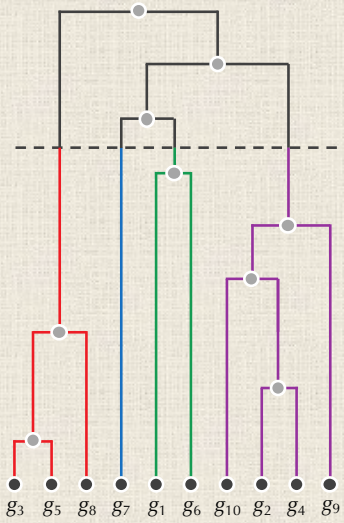
INTRODUCTION

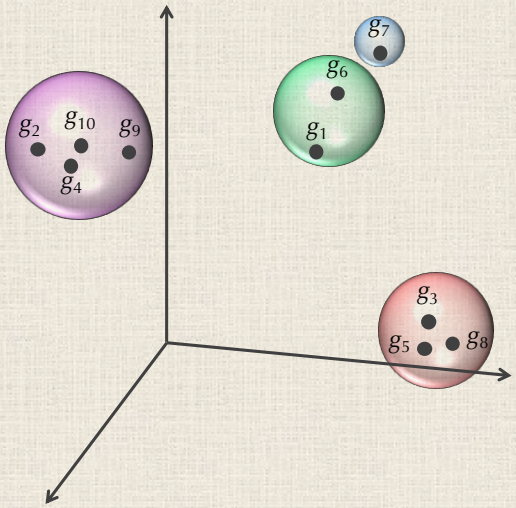
The purpose of the study reported here was to develop a quantitative index of relationship between any two species of a higher systematic unit, as well as to exploit such indices of association in the establishment of a satisfactory hierarchy. The authors became interested in the development of such a method when they attempted to find a technique for classifying organisms that was free from the subjectivity inherent in customary taxonomic procedure.

1. Contribution number 945 from the Department of Entomology, University of Kansas.
2. We wish to acknowledge the constructive criticism received in connection with this and related work from the following individuals who kindly gave their time to read and comment upon the manuscript: Paul R. Ehrlich, University of Kansas; Raymond B. Cattell, University of Illinois; Alfred E. Emerson, University of Chicago; Warwick E. Kerr, Universidade de São Paulo; Ernst Mayr, Harvard University; Louis L. McQuitty, Michigan State University; G. G. Simpson, American Museum of Natural History; Peter C. Silvester-Bradley, University of Kansas and University of Sheffield; and Paulo E. Vanzolini, Departmento de Zoologia, Secretaria de Agricultura, São Paulo. These persons, however, are not responsible for the opinions which we have expressed.
Acknowledgment is also due to the University of Kansas General Research Fund for assistance.

(1409)

In your ~~"ML"~~ AI future, you may learn about **hierarchical clustering**, a generalization of UPGMA.

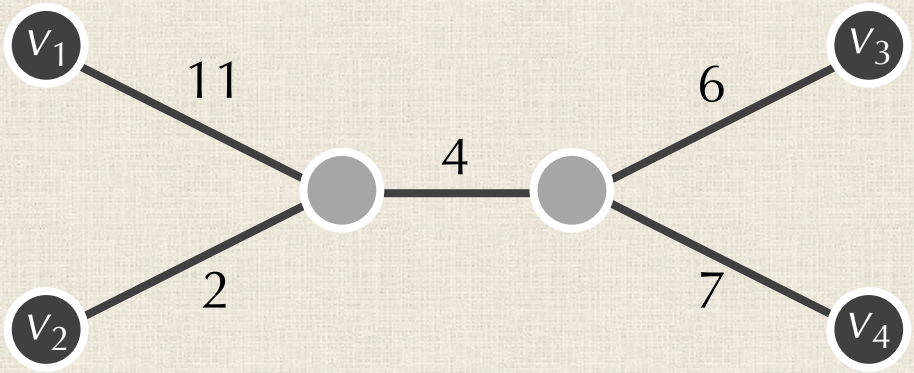


# THE NEIGHBOR-JOINING ALGORITHM

# Recall: Outliers were the Issue

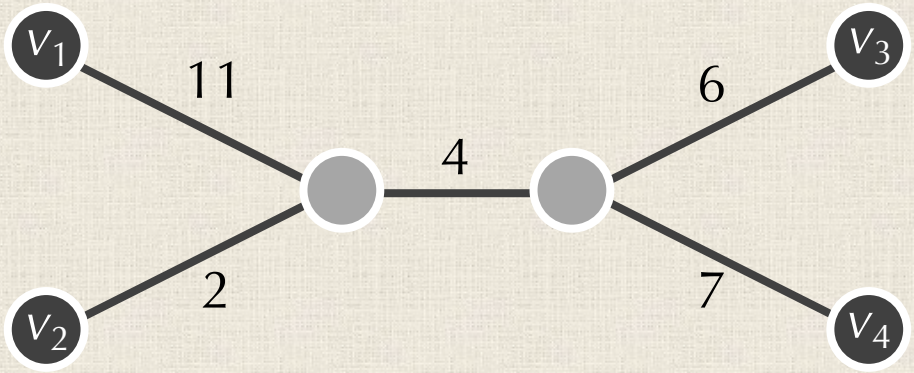
| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |




Recall: Outliers prevented minimum elements from corresponding to neighbors.

# Recall: Outliers were the Issue

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |




Recall: Outliers prevented minimum elements from corresponding to neighbors.

**Key Insight:** What if our idea of joining neighbors was GREAT, we just need to change the matrix?

# The Neighbor-Joining Theorem

Given an $n$ x $n$ distance matrix $D$, its **neighbor-joining matrix** is the matrix $D^*$ defined as

$$D^*_{i,j} = (n - 2) \bullet D_{i,j} - TotalDistance_D(i) - TotalDistance_D(j)$$

where $TotalDistance_D(i)$ is the sum of distances from $i$ to all other leaves.

# The Neighbor-Joining Theorem

Given an $n$ x $n$ distance matrix $D$, its **neighbor-joining matrix** is the matrix $D^*$ defined as

$$D^*_{i,j} = (n - 2) \bullet D_{i,j} - TotalDistance_D(i) - TotalDistance_D(j)$$

where $TotalDistance_D(i)$ is the sum of distances from $i$ to all other leaves.

$D$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |

# The Neighbor-Joining Theorem

Given an $n$ x $n$ distance matrix $D$, its **neighbor-joining matrix** is the matrix $D^*$ defined as

$$D^*_{i,j} = (n - 2) \bullet D_{i,j} - TotalDistance_D(i) - TotalDistance_D(j)$$

where $TotalDistance_D(i)$ is the sum of distances from $i$ to all other leaves.

$D$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |

| $TotalDistance_D$ |
|---|
| 56 |
| 38 |
| 46 |
| 48 |

# The Neighbor-Joining Theorem

Given an $n$ x $n$ distance matrix $D$, its **neighbor-joining matrix** is the matrix $D^*$ defined as

$$D^*_{i,j} = (n - 2) \bullet D_{i,j} - TotalDistance_D(i) - TotalDistance_D(j)$$

where $TotalDistance_D(i)$ is the sum of distances from $i$ to all other leaves.

$D$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |

| $TotalDistance_D$ |
|---|
| 56 |
| 38 |
| 46 |
| 48 |

$D^*$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | -68 | -60 | -60 |
| $v_2$ | -68 | 0 | -60 | -60 |
| $v_3$ | -60 | -60 | 0 | -68 |
| $v_4$ | -60 | -60 | -68 | 0 |

# The Neighbor-Joining Theorem

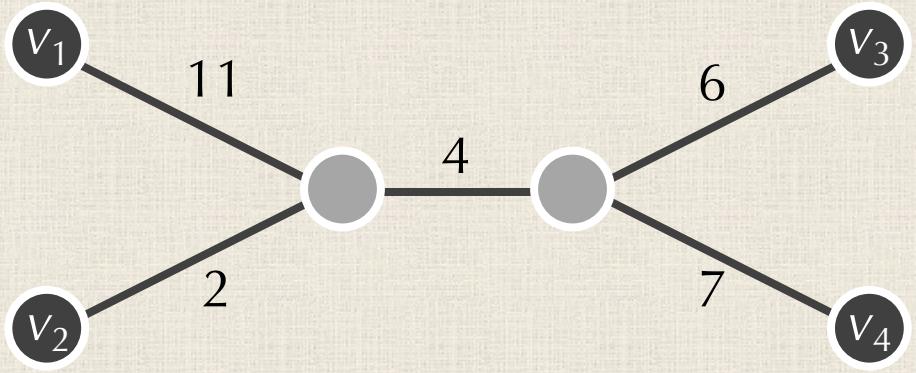
Given an $n$ x $n$ distance matrix $D$, its **neighbor-joining matrix** is the matrix $D^*$ defined as

$$D^*_{i,j} = (n - 2) \bullet D_{i,j} - TotalDistance_D(i) - TotalDistance_D(j)$$

**STOP:** What does $D^*$ do to outliers?

$D$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |



# The Neighbor-Joining Theorem

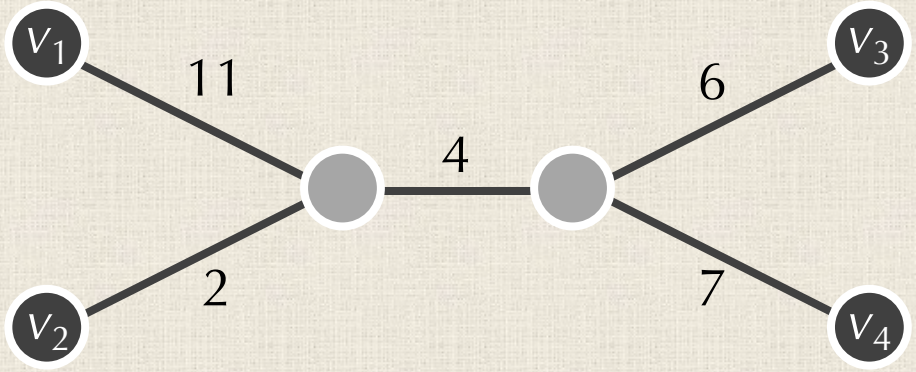
Given an $n$ x $n$ distance matrix $D$, its **neighbor-joining matrix** is the matrix $D^*$ defined as

$$D^*_{i,j} = (n - 2) \bullet D_{i,j} - TotalDistance_D(i) - TotalDistance_D(j)$$

**Answer:** If $i$ is an outlier, $TotalDistance_D(i)$ goes up, which causes $D^*_{i,j}$ to go down.

$D$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |



# The Neighbor-Joining Theorem

**Neighbor-Joining Theorem:** If $D$ is additive, then the smallest element of $D^*$ corresponds to neighboring leaves in $Tree(D)$!

$D$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $TotalDistance_D$ |
|---|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 | 56 |
| $v_2$ | 13 | 0 | 12 | 13 | 38 |
| $v_3$ | 21 | 12 | 0 | 13 | 46 |
| $v_4$ | 22 | 13 | 13 | 0 | 48 |

$D^*$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | -68 | -60 | -60 |
| $v_2$ | -68 | 0 | -60 | -60 |
| $v_3$ | -60 | -60 | 0 | -68 |
| $v_4$ | -60 | -60 | -68 | 0 |

# The Neighbor-Joining Theorem

**Neighbor-Joining Theorem:** If $D$ is additive, then the smallest element of $D^*$ corresponds to neighboring leaves in $Tree(D)$!

$D$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $TotalDistance_D$ |
|---|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 | 56 |
| $v_2$ | 13 | 0 | **12** | 13 | 38 |
| $v_3$ | 21 | **12** | 0 | 13 | 46 |
| $v_4$ | 22 | 13 | 13 | 0 | 48 |

$D^*$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | **-68** | -60 | -60 |
| $v_2$ | **-68** | 0 | -60 | -60 |
| $v_3$ | -60 | -60 | 0 | **-68** |
| $v_4$ | -60 | -60 | **-68** | 0 |

# Neighbor-Joining in Action

$D^*$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $TotalDistance_D$ |
|---|---|---|---|---|---|
| $v_1$ | 0 | -68 | -60 | -60 | 56 |
| $v_2$ | -68 | 0 | -60 | -60 | 38 |
| $v_3$ | -60 | -60 | 0 | -68 | 46 |
| $v_4$ | -60 | -60 | -68 | 0 | 48 |

1. Construct neighbor-joining matrix $D^*$ from $D$.

# Neighbor-Joining in Action

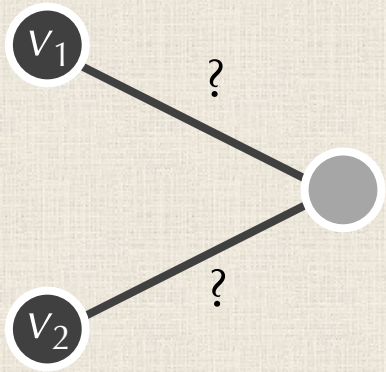
| | | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $TotalDistance_D$ |
|---|---|---|---|---|---|---|
| | $v_1$ | 0 | **-68** | -60 | -60 | 56 |
| $D^*$ | $v_2$ | **-68** | 0 | -60 | -60 | 38 |
| | $v_3$ | -60 | -60 | 0 | **-68** | 46 |
| | $v_4$ | -60 | -60 | **-68** | 0 | 48 |

2. Find a minimum element $D^*_{i,j}$ of $D^*$.

# Neighbor-Joining in Action

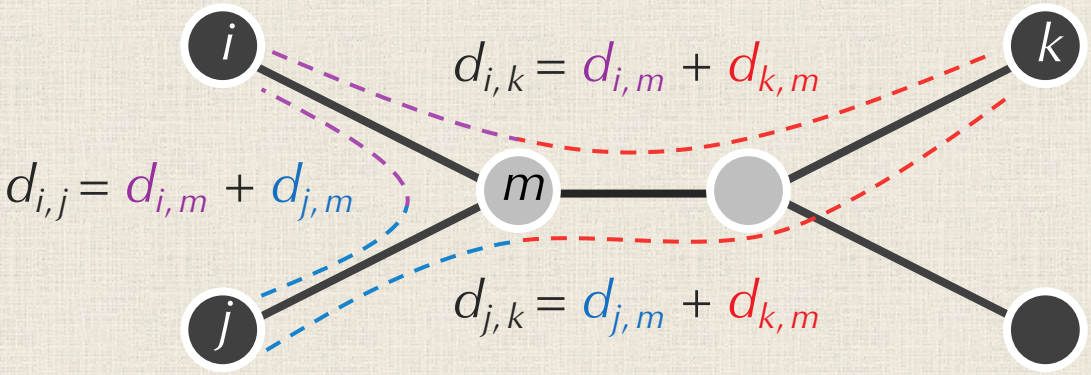
$D^*$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $TotalDistance_D$ |
|---|---|---|---|---|---|
| $v_1$ | 0 | **-68** | -60 | -60 | 56 |
| $v_2$ | **-68** | 0 | -60 | -60 | 38 |
| $v_3$ | -60 | -60 | 0 | -68 | 46 |
| $v_4$ | -60 | -60 | -68 | 0 | 48 |




By the Neighbor-Joining Theorem, we know that this means that $v_1$ and $v_2$ are neighbors. But what are the "?" distances?

# Recall: Computing Length of a Limb




$$d_{k,m} = [(d_{i,m} + d_{k,m}) + (d_{j,m} + d_{k,m}) - (d_{i,m} + d_{j,m})] \,/\, 2$$

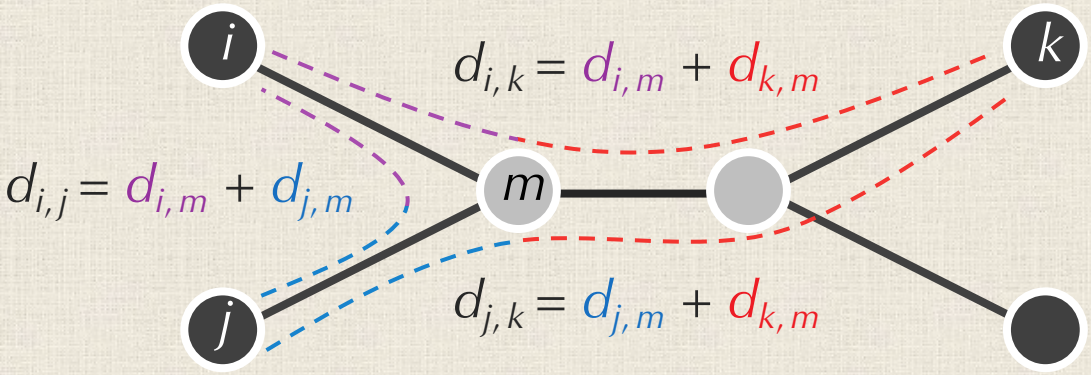
$$d_{k,m} = (d_{i,k} + d_{j,k} - d_{i,j}) \,/\, 2$$

$$d_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j}) \,/\, 2$$

$$\therefore \; d_{i,m} = D_{i,k} - (D_{i,k} + D_{j,k} - D_{i,j}) \,/\, 2$$

$$d_{i,m} = (D_{i,k} + D_{i,j} - D_{j,k}) \,/\, 2$$

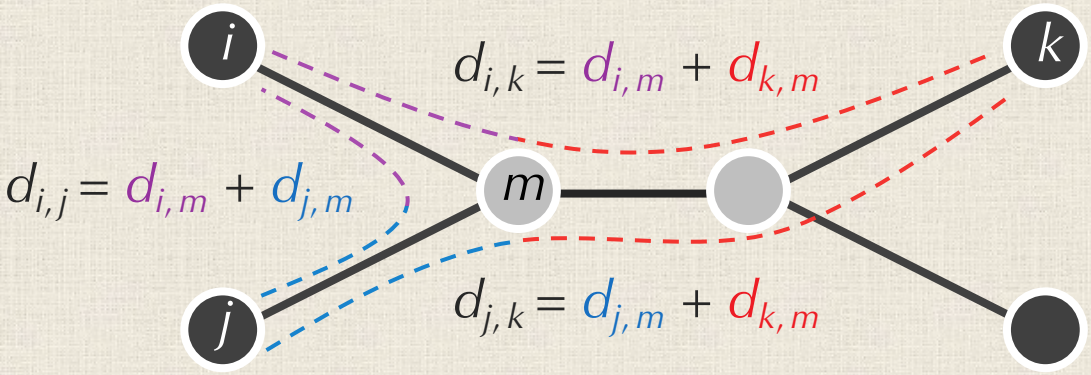
# Recall: Computing Length of a Limb




Define a **limb** as the edge from a leaf to its parent, and *LimbLength*(*i*) as the length of the limb at leaf *i*.

$$d_{i,m} = (D_{i,k} + D_{i,j} - D_{j,k}) / 2 = LimbLength(i)$$

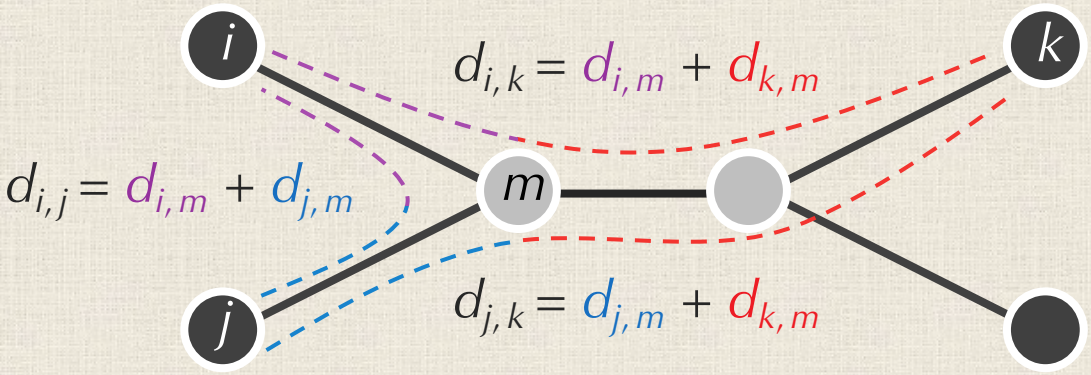
# Recall: Computing Length of a Limb




**STOP:** The following formula must hold if $D$ is additive and $i$ and $j$ are neighbors ... But what if $D$ is not additive?

$$d_{i,m} = (D_{i,k} + D_{i,j} - D_{j,k}) / 2 = LimbLength(i)$$

# Recall: Computing Length of a Limb



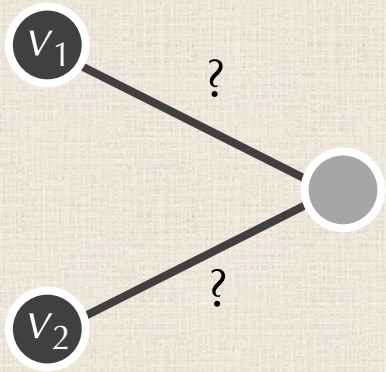

**Answer:** Take the *average* of this formula over every $k$ other than $i$ and $j$.

$$d_{i,m} = (D_{i,k} + D_{i,j} - D_{j,k}) / 2 = LimbLength(i)$$

# Neighbor-Joining in Action

$D^*$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $TotalDistance_D$ |
|---|---|---|---|---|---|
| $v_1$ | 0 | **-68** | -60 | -60 | **56** |
| $v_2$ | **-68** | 0 | -60 | -60 | **38** |
| $v_3$ | -60 | -60 | 0 | -68 | 46 |
| $v_4$ | -60 | -60 | -68 | 0 | 48 |

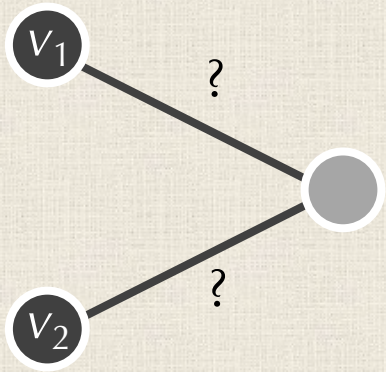



3. Having found that *i* and *j* are neighbors, take the average $(D_{i,k} + D_{i,j} - D_{j,k})/2$ over all *k* not equal to *i*, *j*. Set *LimbLength*(*i*) equal to this average.

# Neighbor-Joining in Action

$D$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |



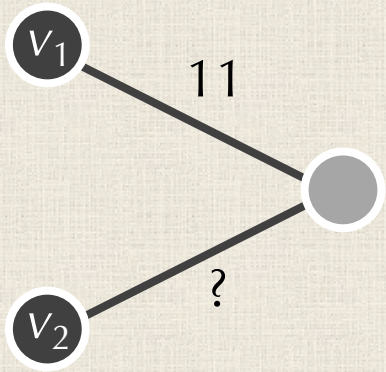

**Exercise:** What should be the limb length of $v_1$?

3. Having found that $i$ and $j$ are neighbors, take the average $(D_{i,k} + D_{i,j} - D_{j,k})/2$ over all $k$ not equal to $i$, $j$. Set *LimbLength*($i$) equal to this average.

# Neighbor-Joining in Action

$D$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |

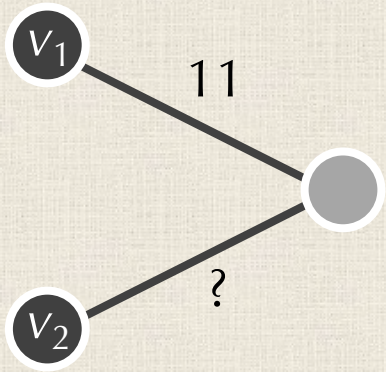



4. Set *LimbLength*(*j*) equal to $D_{i,j}$ – *LimbLength*(*i*).

# Neighbor-Joining in Action

$D$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |



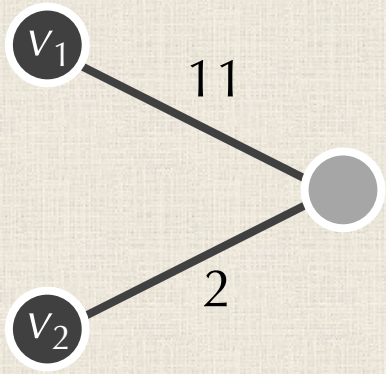

**Exercise:** What should be the limb length of $v_2$?

4. Set *LimbLength*($j$) equal to $D_{i,j}$ – *LimbLength*($i$).

# Neighbor-Joining in Action

$D$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |

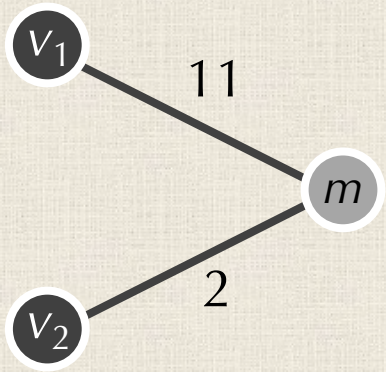



4. Set *LimbLength*(*j*) equal to $D_{i,j}$ – *LimbLength*(*i*).

# Neighbor-Joining in Action

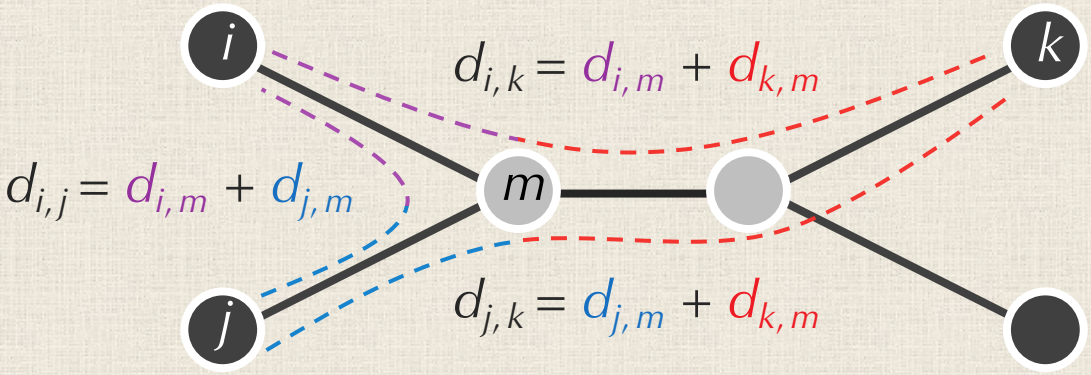
$D$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |




Now we want to apply a recursive algorithm. To do so, we ask what the distance from *m* (the parent of our neighbors) to every other node should be.

# Flashback: Computation of $d_{k,m}$




$$d_{k,m} = [(d_{i,m} + d_{k,m}) + (d_{j,m} + d_{k,m}) - (d_{i,m} + d_{j,m})] / 2$$

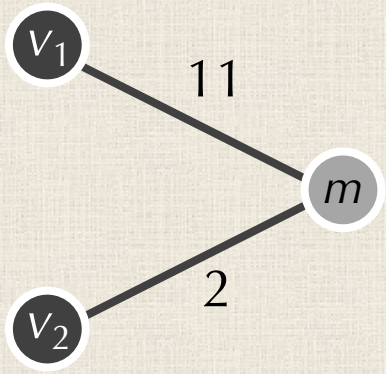
$$d_{k,m} = (d_{i,k} + d_{j,k} - d_{i,j}) / 2$$

$$d_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j}) / 2$$

# Neighbor-Joining in Action

$D$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |



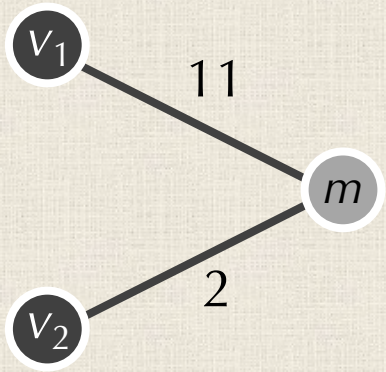

5. Form a matrix $D'$ by removing $i$-th and $j$-th row/column from $D$ and adding an $m$-th row/column such that for any $k$, $D'_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j}) / 2$.

# Neighbor-Joining in Action

**Exercise:** Compute distance from $m$ to $v_3$ and $v_4$ .

$D$

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ |
|---|---|---|---|---|
| $v_1$ | 0 | 13 | 21 | 22 |
| $v_2$ | 13 | 0 | 12 | 13 |
| $v_3$ | 21 | 12 | 0 | 13 |
| $v_4$ | 22 | 13 | 13 | 0 |

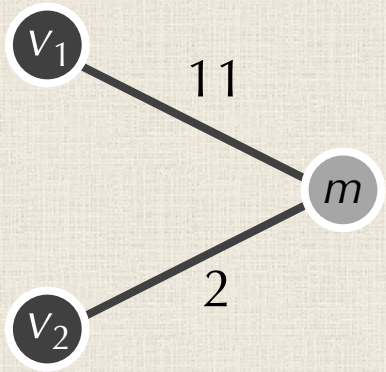



5. Form a matrix $D'$ by removing $i$-th and $j$-th row/column from $D$ and adding an $m$-th row/column such that for any $k$, $D'_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j}) / 2$.

# Neighbor-Joining in Action

$D'$

| | $m$ | $v_3$ | $v_4$ |
|---|---|---|---|
| $m$ | 0 | 10 | 11 |
| $v_3$ | 10 | 0 | 13 |
| $v_4$ | 11 | 13 | 0 |

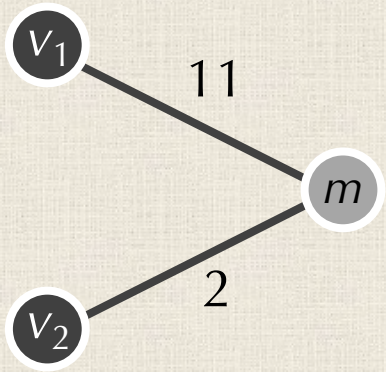



5. Form a matrix $D'$ by removing $i$-th and $j$-th row/column from $D$ and adding an $m$-th row/column such that for any $k$, $D'_{k,m} = (D_{i,k} + D_{j,k} - D_{i,j}) / 2$.

# Neighbor-Joining in Action

$D'$

| | $m$ | $v_3$ | $v_4$ |
|---|---|---|---|
| $m$ | 0 | 10 | 11 |
| $v_3$ | 10 | 0 | 13 |
| $v_4$ | 11 | 13 | 0 |

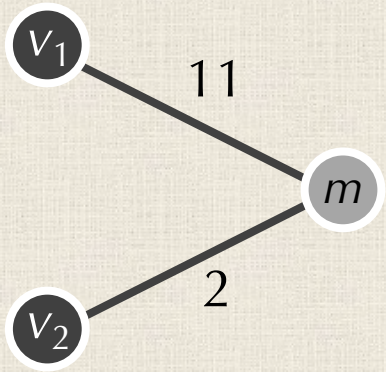



**STOP:** What should we do now?

# Neighbor-Joining in Action

$D'$

| | $m$ | $v_3$ | $v_4$ |
|---|---|---|---|
| $m$ | 0 | 10 | 11 |
| $v_3$ | 10 | 0 | 13 |
| $v_4$ | 11 | 13 | 0 |




**STOP:** What should we do now?

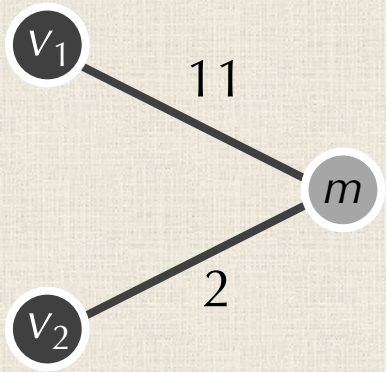
**Answer:** Recursion on our 3 x 3 matrix!

# Neighbor-Joining in Action

**Exercise:** Carry out one more step of the algorithm.

$D'$

| | $m$ | $v_3$ | $v_4$ |
|---|---|---|---|
| $m$ | 0 | 10 | 11 |
| $v_3$ | 10 | 0 | 13 |
| $v_4$ | 11 | 13 | 0 |

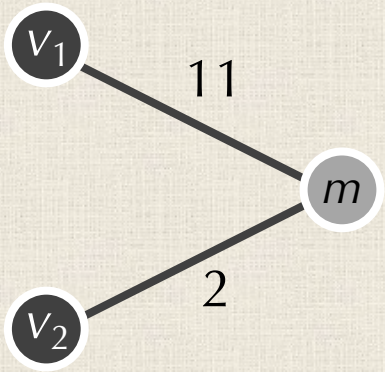



1. Construct neighbor-joining matrix $D^*$ from $D$.
2. Find a minimum element $D^*_{i,j}$ of $D^*$.
3. Having found that $i$ and $j$ are neighbors, take the average $(D_{i,k} + D_{i,j} - D_{j,k}) / 2$ over all $k$ not equal to $i$, $j$. Set LimbLength($i$) equal to this average.
4. Set LimbLength($j$) equal to $D_{i,j}$ – *LimbLength*($j$).
5. Form a matrix $D'$ by removing $i$-th and $j$-th row/column from $D$ and adding an $m$-th row/column such that for any $k$, $D_{k,m} = (D_{k,i} + D_{k,j} - D_{i,j}) / 2$.

# Neighbor-Joining in Action

$D'$

| | $m$ | $v_3$ | $v_4$ | $TotalDistance_{D'}$ |
|---|---|---|---|---|
| $m$ | 0 | 10 | 11 | 21 |
| $v_3$ | 10 | 0 | 13 | 23 |
| $v_4$ | 11 | 13 | 0 | 24 |




$D'^*$

| | $m$ | $v_3$ | $v_4$ |
|---|---|---|---|
| $m$ | 0 | -34 | -34 |
| $v_3$ | -34 | 0 | -34 |
| $v_4$ | -34 | -34 | 0 |

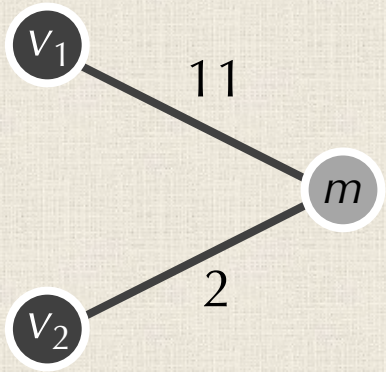
1. Construct neighbor-joining matrix.

# Neighbor-Joining in Action

$D'$

| | $m$ | $v_3$ | $v_4$ |
|---|---|---|---|
| $m$ | 0 | 10 | 11 |
| $v_3$ | 10 | 0 | 13 |
| $v_4$ | 11 | 13 | 0 |

| $TotalDistance_{D'}$ |
|---|
| 21 |
| 23 |
| 24 |




$D'^*$

| | $m$ | $v_3$ | $v_4$ |
|---|---|---|---|
| $m$ | 0 | -34 | -34 |
| $v_3$ | -34 | 0 | -34 |
| $v_4$ | -34 | -34 | 0 |

They're all neighbors!

2. Find a minimum element $D^*_{i,j}$ of $D^*$.

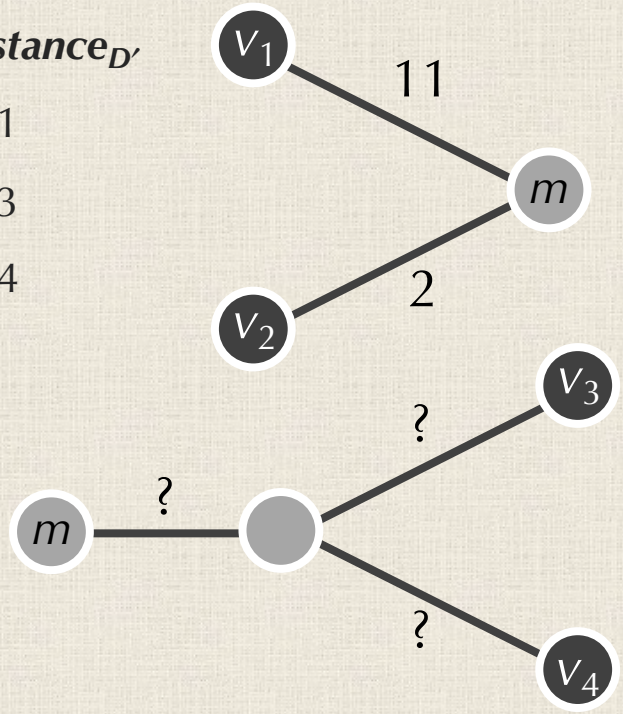
# Neighbor-Joining in Action

$D'$

| | $m$ | $v_3$ | $v_4$ | $TotalDistance_{D'}$ |
|---|---|---|---|---|
| $m$ | 0 | 10 | 11 | 21 |
| $v_3$ | 10 | 0 | 13 | 23 |
| $v_4$ | 11 | 13 | 0 | 24 |

$D'^*$

| | $m$ | $v_3$ | $v_4$ |
|---|---|---|---|
| $m$ | 0 | -34 | -34 |
| $v_3$ | -34 | 0 | -34 |
| $v_4$ | -34 | -34 | 0 |




3. For any $i$, set $LimbLength(i)$ equal to the average $(D_{i,k} + D_{i,j} - D_{j,k})/2$ over all $k$ not equal to $i$, $j$.

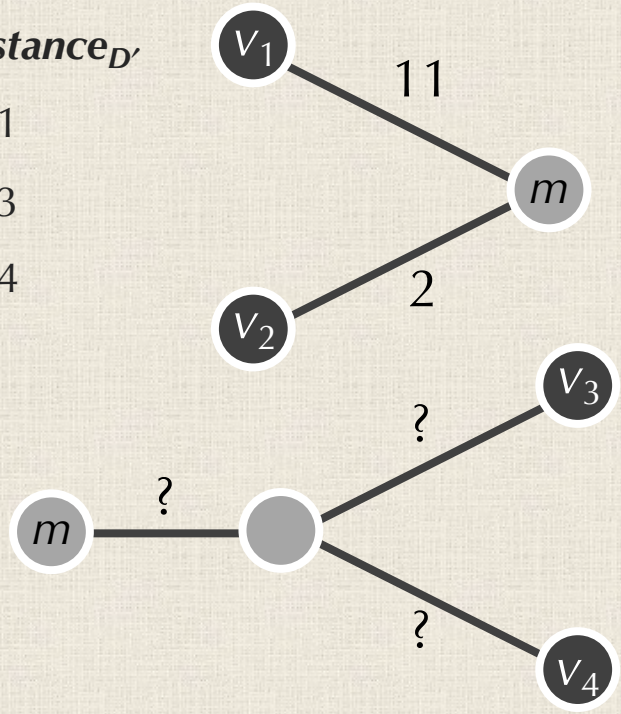
# Neighbor-Joining in Action

$D'$

| | $m$ | $v_3$ | $v_4$ | $TotalDistance_{D'}$ |
|---|---|---|---|---|
| $m$ | 0 | 10 | 11 | 21 |
| $v_3$ | 10 | 0 | 13 | 23 |
| $v_4$ | 11 | 13 | 0 | 24 |

$D'^*$

| | $m$ | $v_3$ | $v_4$ |
|---|---|---|---|
| $m$ | 0 | -34 | -34 |
| $v_3$ | -34 | 0 | -34 |
| $v_4$ | -34 | -34 | 0 |




**STOP:** What are the limb lengths?

# Neighbor-Joining in Action

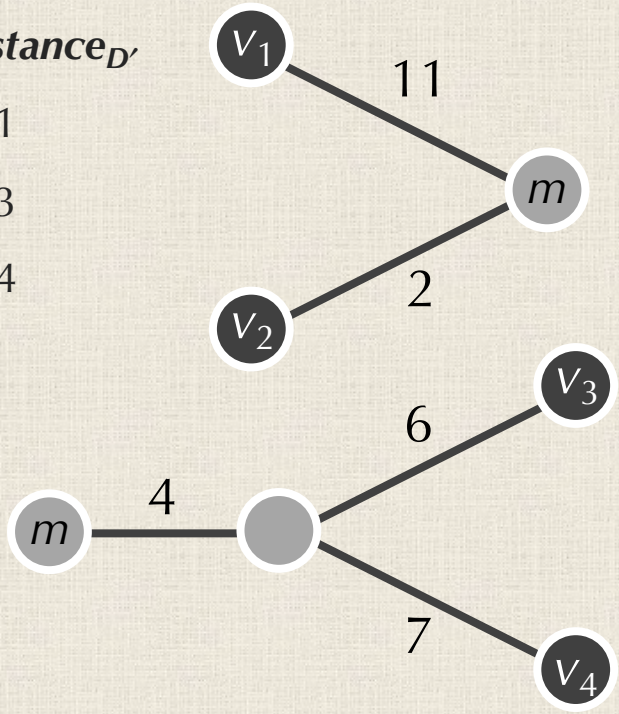
$D'$

| | $m$ | $v_3$ | $v_4$ |
|---|---|---|---|
| $m$ | 0 | 10 | 11 |
| $v_3$ | 10 | 0 | 13 |
| $v_4$ | 11 | 13 | 0 |

| $TotalDistance_{D'}$ |
|---|
| 21 |
| 23 |
| 24 |

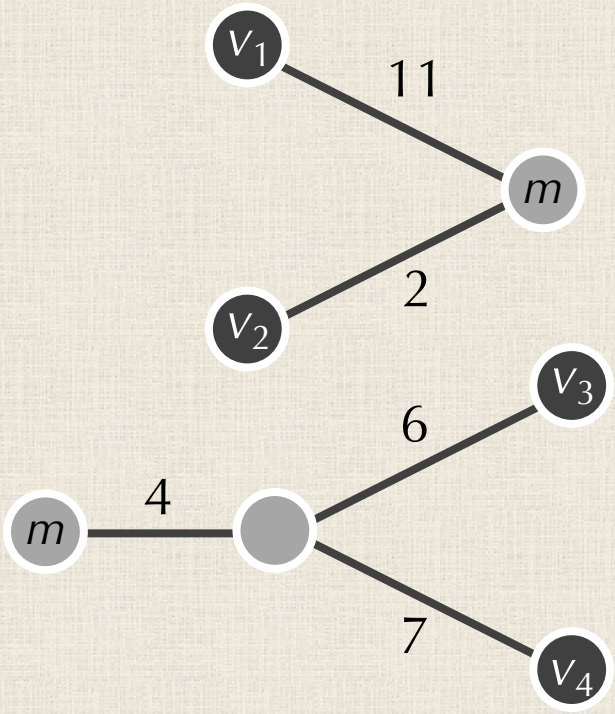
$D'^*$

| | $m$ | $v_3$ | $v_4$ |
|---|---|---|---|
| $m$ | 0 | -34 | -34 |
| $v_3$ | -34 | 0 | -34 |
| $v_4$ | -34 | -34 | 0 |



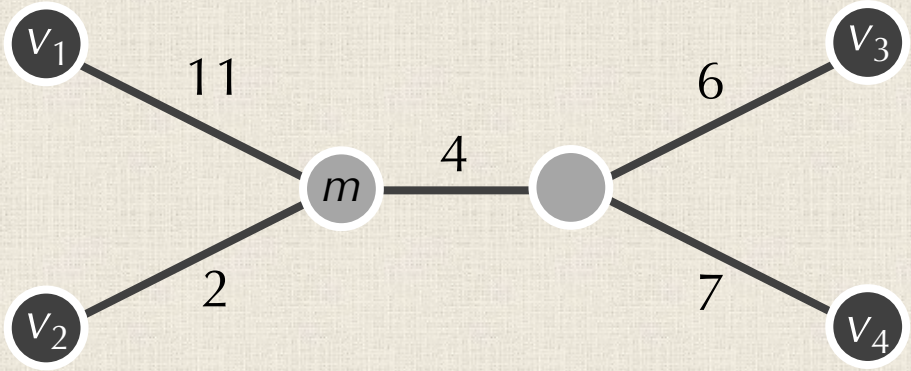

We hit a base case! Now add remaining limbs …

# Neighbor-Joining in Action




We hit a base case! Now add remaining limbs …

# Neighbor-Joining in Action




… and we're done!

# Neighbor-Joining Summary

**NeighborJoining(*D*):**

1. Construct neighbor-joining matrix $D^*$ from $D$.
2. Find a minimum element $D^*_{i,j}$ of $D^*$.
3. Having found that $i$ and $j$ are neighbors, take the average $(D_{i,k} + D_{i,j} - D_{j,k})$ over all $k$ not equal to $i$, $j$. Set LimbLength($i$) equal to this average.
4. Set LimbLength($j$) equal to $D_{i,j} - LimbLength(j)$.
5. Form a matrix $D'$ by removing $i$-th and $j$-th row/column from $D$ and adding an $m$-th row/column such that for any $k$, $D_{k,m} = (D_{k,i} + D_{k,j} - D_{i,j}) / 2$.
6. Apply **NeighborJoining** recursively to $D'$ to obtain $Tree(D')$.
7. Reattach limbs of $i$ and $j$ to obtain $Tree(D)$.

# Weakness of Distance-Based Methods

Distance-based algorithms for evolutionary tree reconstruction say nothing about ancestral states at internal nodes.

# Weakness of Distance-Based Methods

Distance-based algorithms for evolutionary tree reconstruction say nothing about ancestral states at internal nodes.

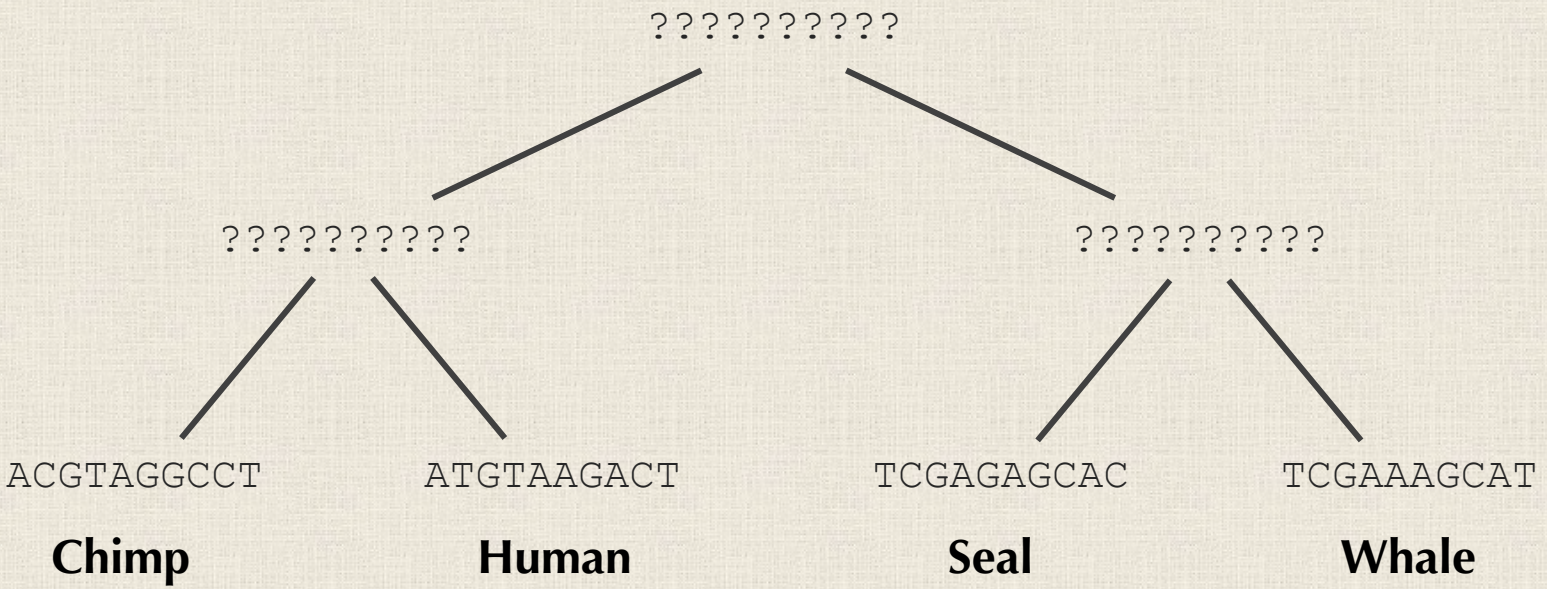
We *lost* information when we converted a multiple alignment to a distance matrix...

| Species | Alignment | Distance Matrix | | | |
|---|---|---|---|---|---|
| | | **Chimp** | **Human** | **Seal** | **Whale** |
| **Chimp** | ACGTAGGCCT | 0 | 3 | 6 | 4 |
| **Human** | ATGTAAGACT | 3 | 0 | 7 | 5 |
| **Seal** | TCGAGAGCAC | 6 | 7 | 0 | 2 |
| **Whale** | TCGAAAGCAT | 4 | 5 | 2 | 0 |

# THE SMALL PARSIMONY ALGORITHM

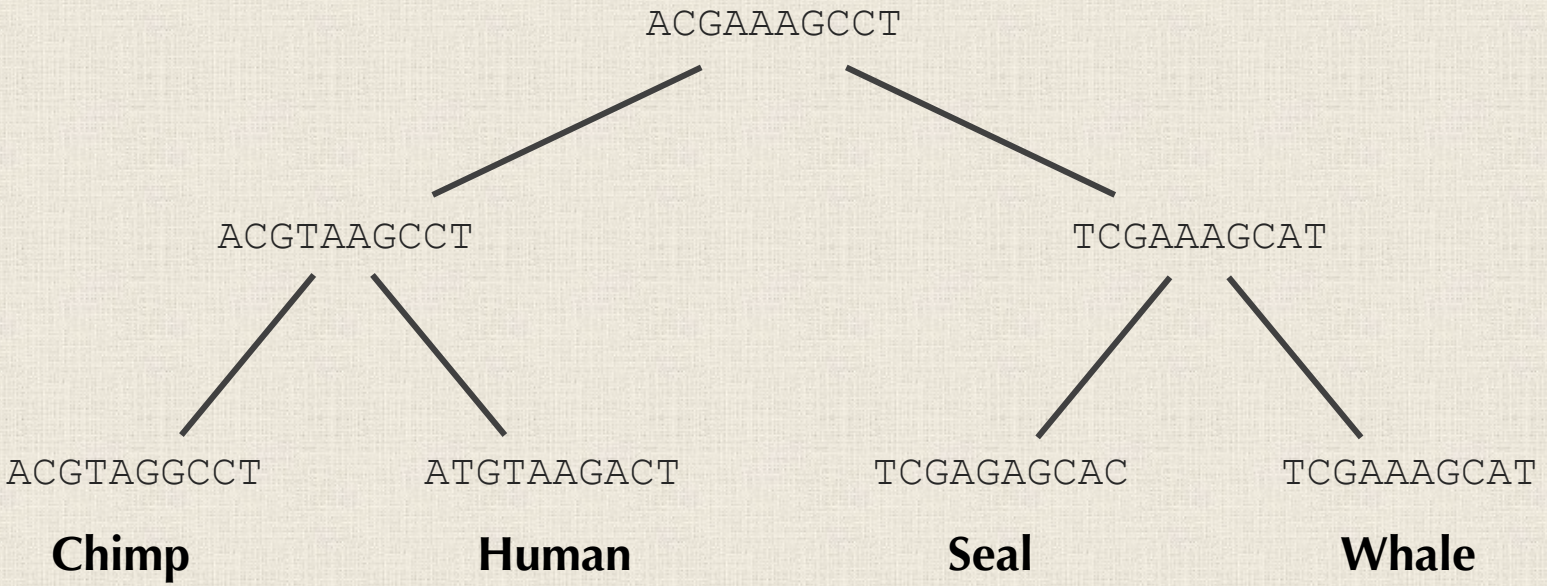
# Goal: Infer Ancestral Sequences

| | |
|---|---|
| **Chimp** | ACGTAGGCCT |
| **Human** | ATGTAAGACT |
| **Seal** | TCGAGAGCAC |
| **Whale** | TCGAAAGCAT |



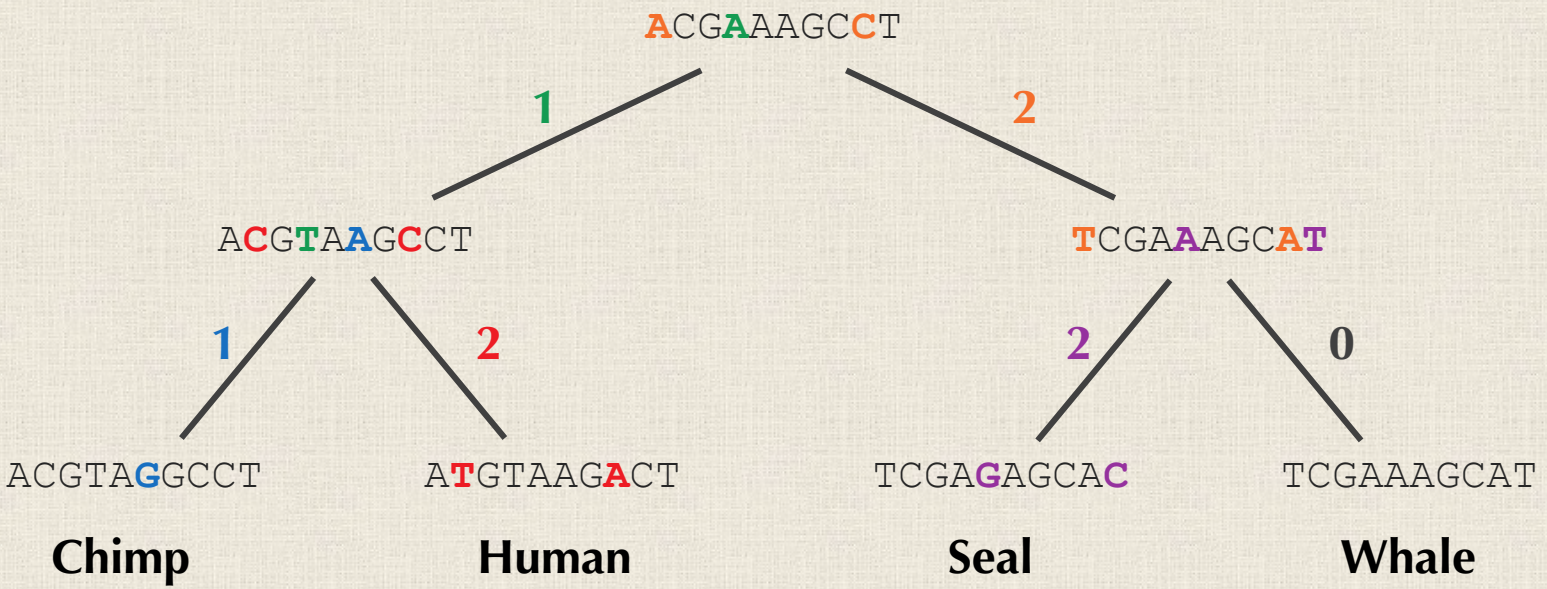
# Goal: Infer Ancestral Sequences

**STOP:** Here's a hypothetical assignment of strings to ancestral nodes. How can we know how good it is?



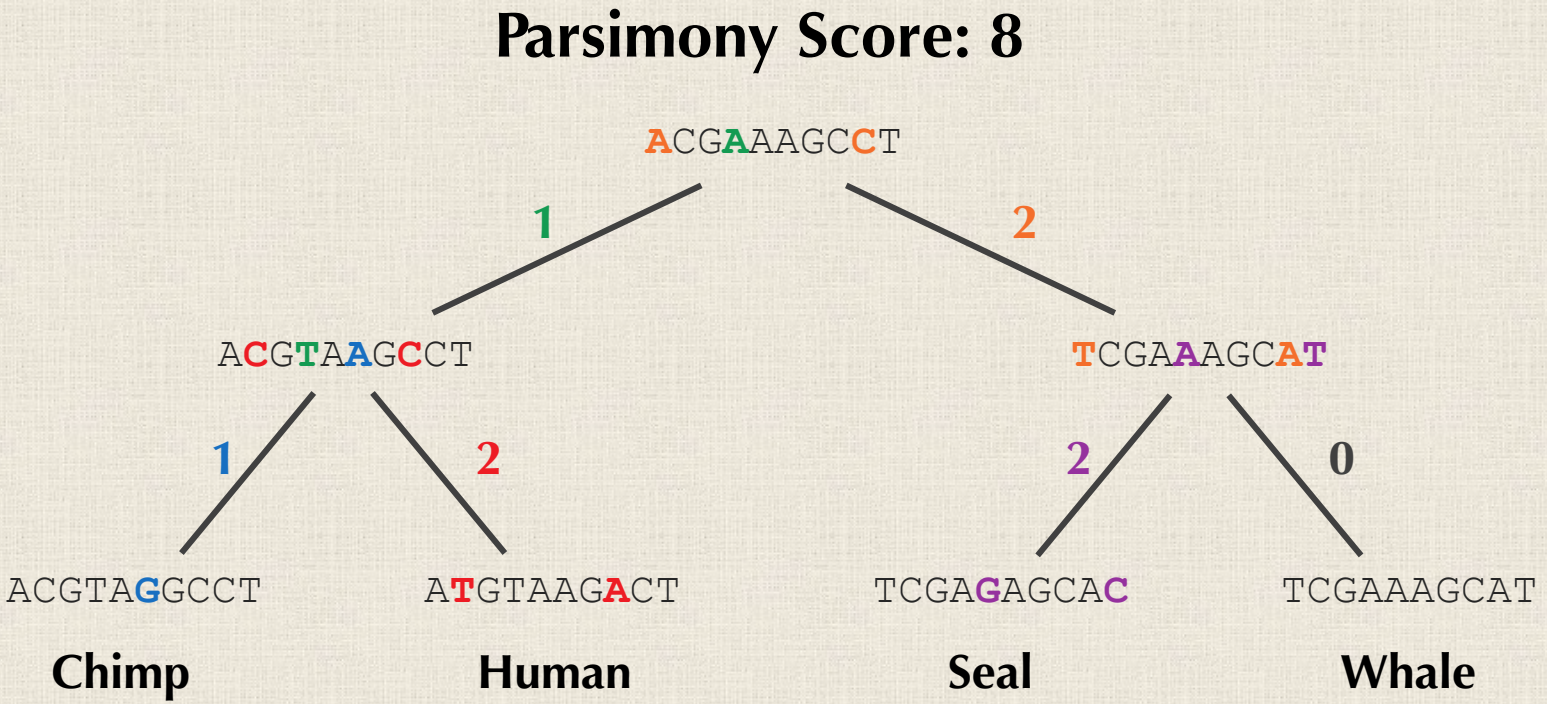
# How good is a *given* assignment of strings to internal nodes?

**Parsimony score:** sum of Hamming distances (total mismatches) along each edge.



# How good is a *given* assignment of strings to internal nodes?

**Parsimony score:** sum of Hamming distances (total mismatches) along each edge.



# What Does "Parsimony" Mean?

**parsimony** noun

par·si·mo·ny | \ ˈpär-sə-ˌmō-nē \

**Definition of *parsimony***

1 **a** : the quality of being careful with money or resources
: THRIFT
// the necessity of wartime *parsimony*

**b** : the quality or state of being stingy
// The charity was surprised by the *parsimony* of some larger corporations.

2 : economy in the use of means to an end

*especially* : economy of explanation in conformity with Occam's razor
// the scientific law of *parsimony* dictates that any example of animal behavior should be interpreted at its simplest, most immediate level
— Peter Gorner


Ockham chooses a razor

# Toward a Computational Problem

**Small Parsimony Problem:**

- **Input:** A rooted binary tree with each leaf labeled by a string of length *m*.
- **Output:** A labeling of all other nodes of the tree by strings of length *m* that minimizes the tree's parsimony score.

# Toward a Computational Problem

**Small Parsimony Problem:**

- **Input:** A rooted binary tree with each leaf labeled by a string of length *m*.
- **Output:** A labeling of all other nodes of the tree by strings of length *m* that minimizes the tree's parsimony score.

**STOP:** Is there any way we can simplify this problem statement?

# Toward a Computational Problem

**Small Parsimony Problem:**

- **Input:** A rooted binary tree with each leaf labeled by a **single symbol**.
- **Output:** A labeling of all other nodes of the tree by **single symbols** that minimizes the tree's parsimony score.

**STOP:** Why is this an acceptable simplification?

# Toward a Computational Problem

**Small Parsimony Problem:**

- **Input:** A rooted binary tree with each leaf labeled by a **single symbol**.
- **Output:** A labeling of all other nodes of the tree by **single symbols** that minimizes the tree's parsimony score.

**Answer:** We may choose to assume that the characters are *independent*.

# Toward a Computational Problem

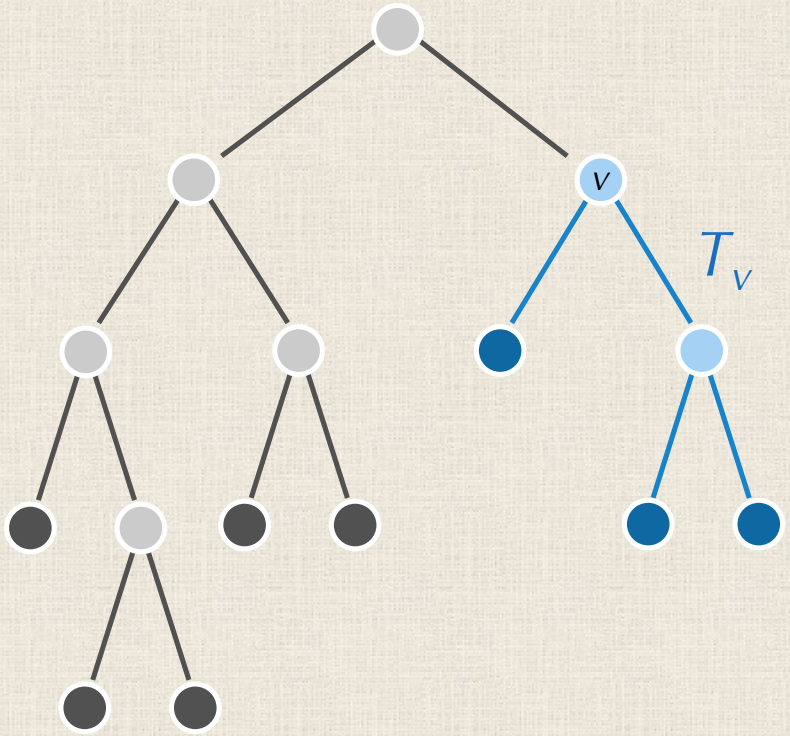
**Small Parsimony Problem:**

- **Input:** A rooted binary tree with each leaf labeled by a **single symbol**.
- **Output:** A labeling of all other nodes of the tree by **single symbols** that minimizes the tree's parsimony score.

**STOP:** Any thoughts on what approach we might use to solve this problem?

# A Dynamic Programming Algorithm

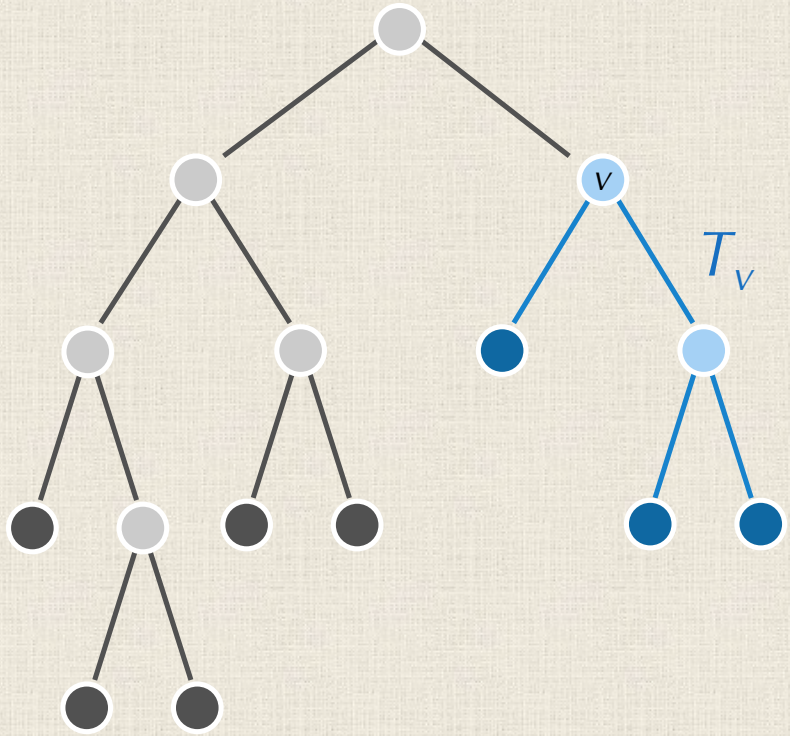
Let $T_v$ denote the subtree of $T$ whose root is $v$.



# A Dynamic Programming Algorithm

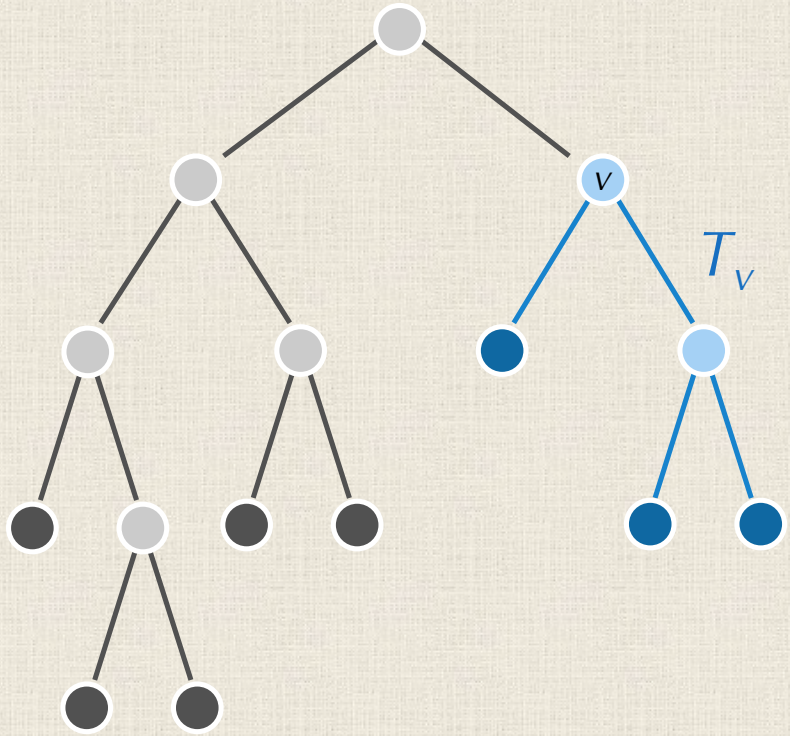
Let $T_v$ denote the subtree of $T$ whose root is $v$.

Define $s_k(v)$ as the minimum parsimony score of $T_v$ over all labelings of $T_v$, assuming that $v$ is labeled by $k$.



# A Dynamic Programming Algorithm

Let $T_v$ denote the subtree of $T$ whose root is $v$.

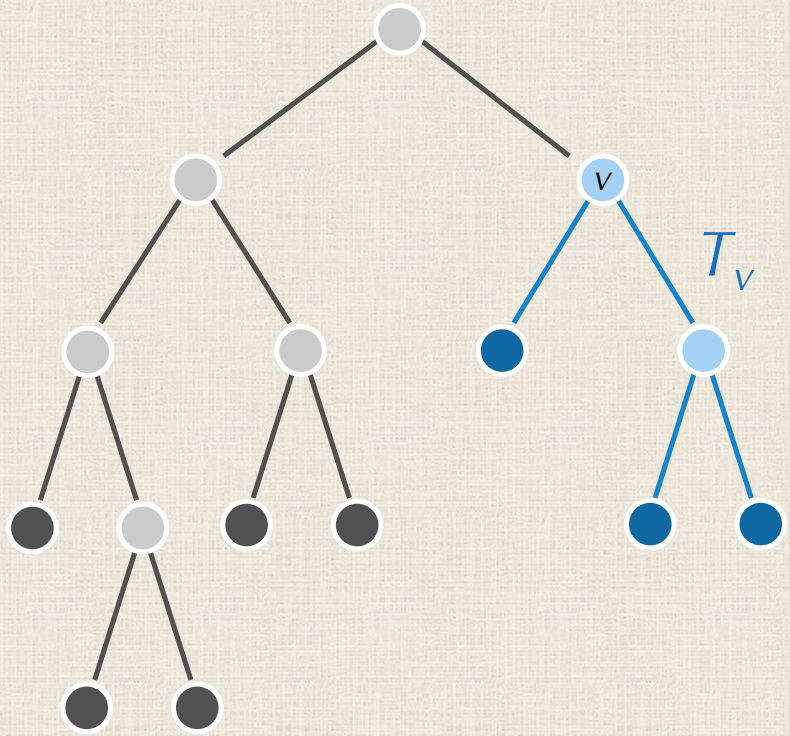



Define $s_k(v)$ as the minimum parsimony score of $T_v$ over all labelings of $T_v$, assuming that $v$ is labeled by $k$.

The minimum parsimony score for the tree is equal to the minimum value of $s_k(root)$ over all symbols $k$.

# A Dynamic Programming Algorithm

Let $T_v$ denote the subtree of $T$ whose root is $v$.




Define $s_k(v)$ as the minimum parsimony score of $T_v$ over all labelings of $T_v$, assuming that $v$ is labeled by $k$.

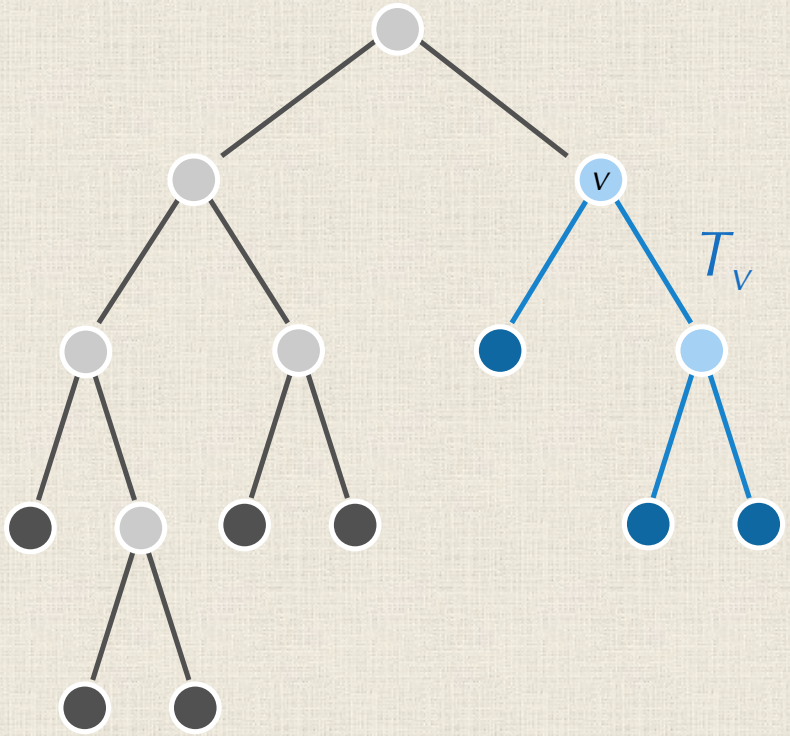
**STOP:** Can you find a recurrence relation for $s_k(v)$?

# A Dynamic Programming Algorithm

For symbols $i$ and $j$, define

- $\alpha_{i,j} = 0$ if $i = j$
- $\alpha_{i,j} = 1$ otherwise.

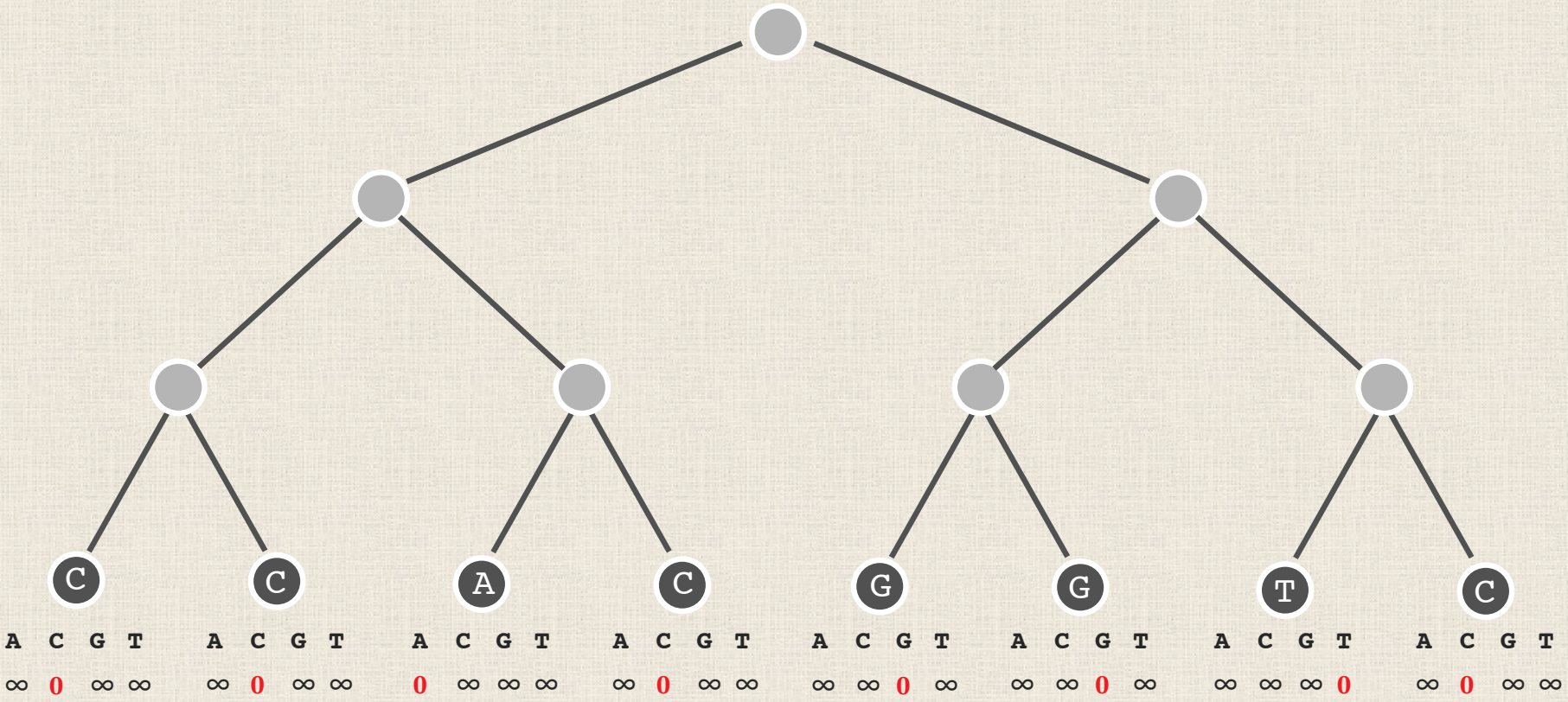
$\alpha$ says, "do they match?"




**Theorem:** The following recurrence relation holds:

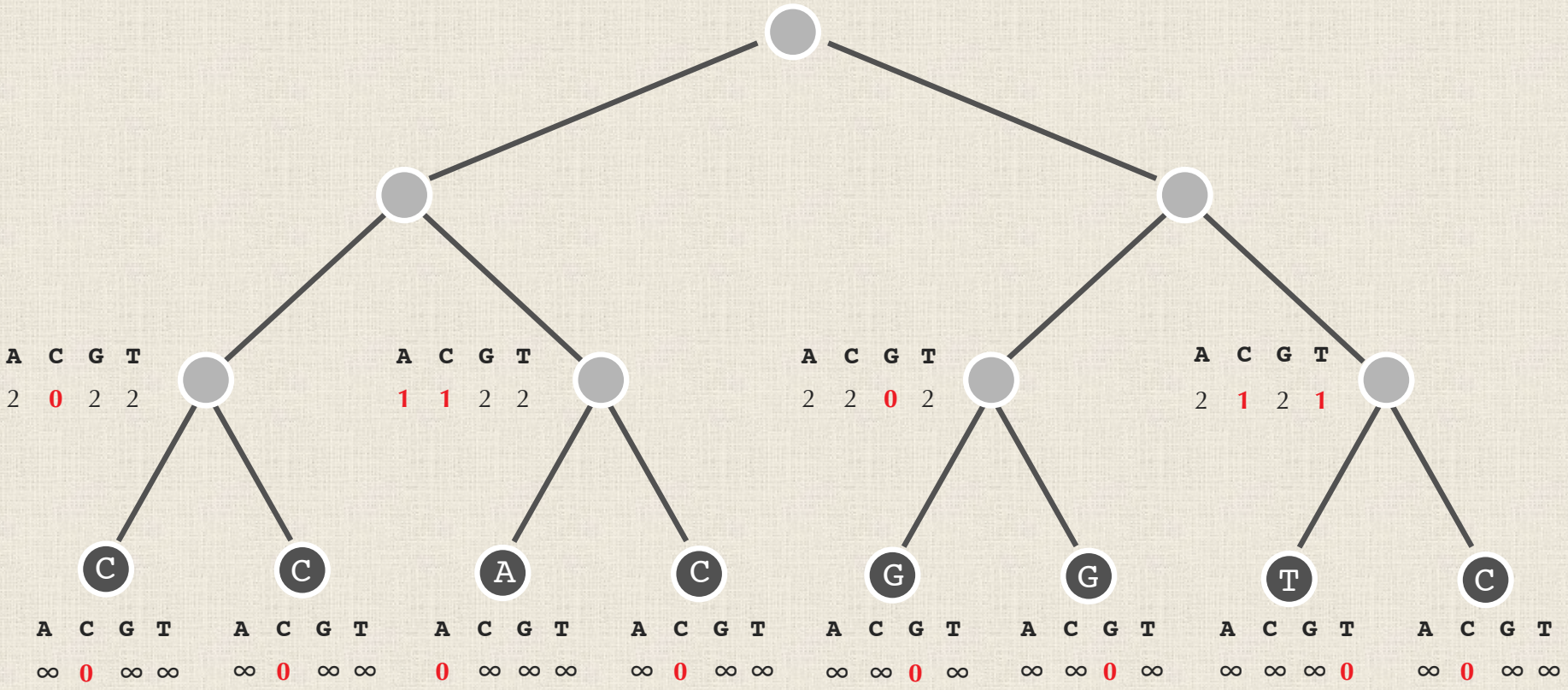
$$s_k(v) = \min_{\text{all symbols } i} \{s_i(Daughter(v)) + \alpha_{i,k}\} + \min_{\text{all symbols } i} \{s_i(Son(v)) + \alpha_{i,k}\}$$

# A Dynamic Programming Algorithm



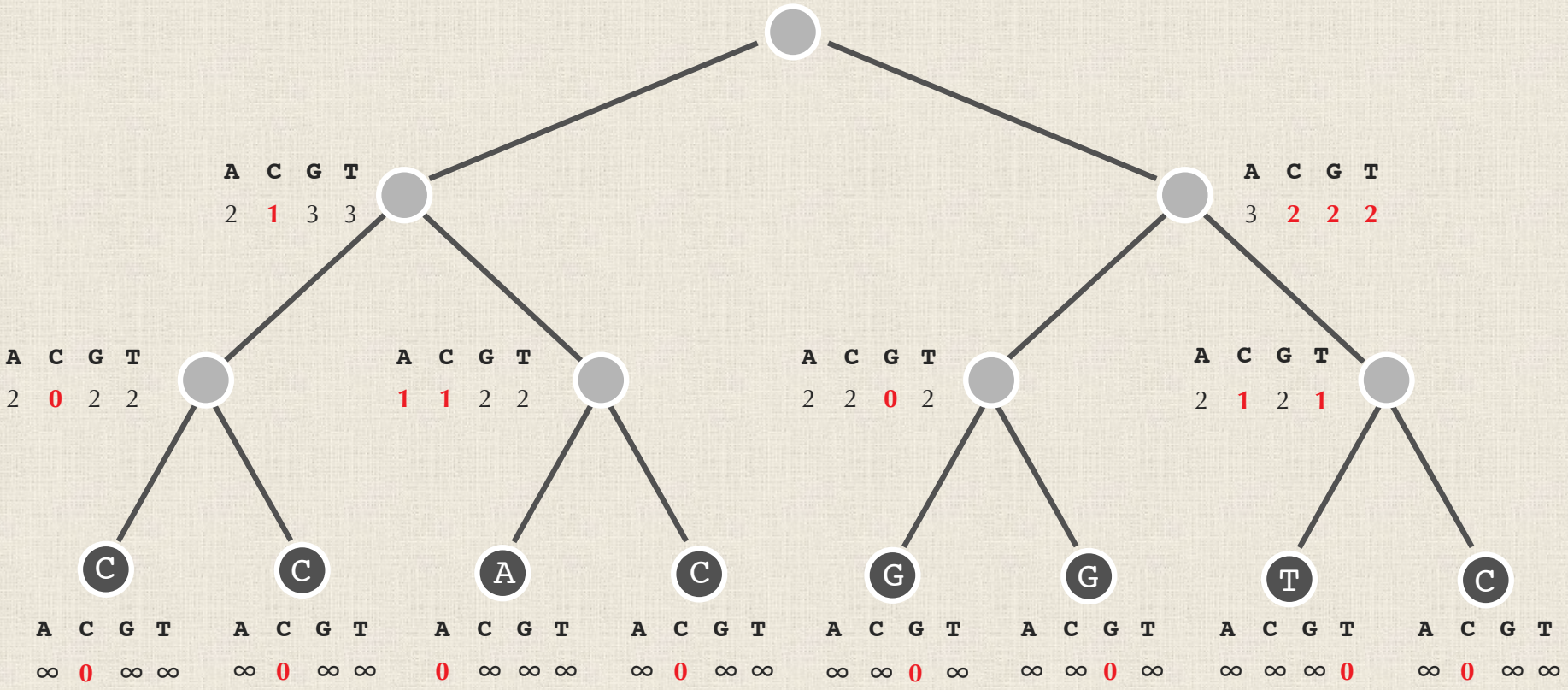

$$s_k(v) = \min_{\text{all symbols } i} \{s_i(Daughter(v)) + \alpha_{i,k}\} + \min_{\text{all symbols } i} \{s_i(Son(v)) + \alpha_{i,k}\}$$

# A Dynamic Programming Algorithm



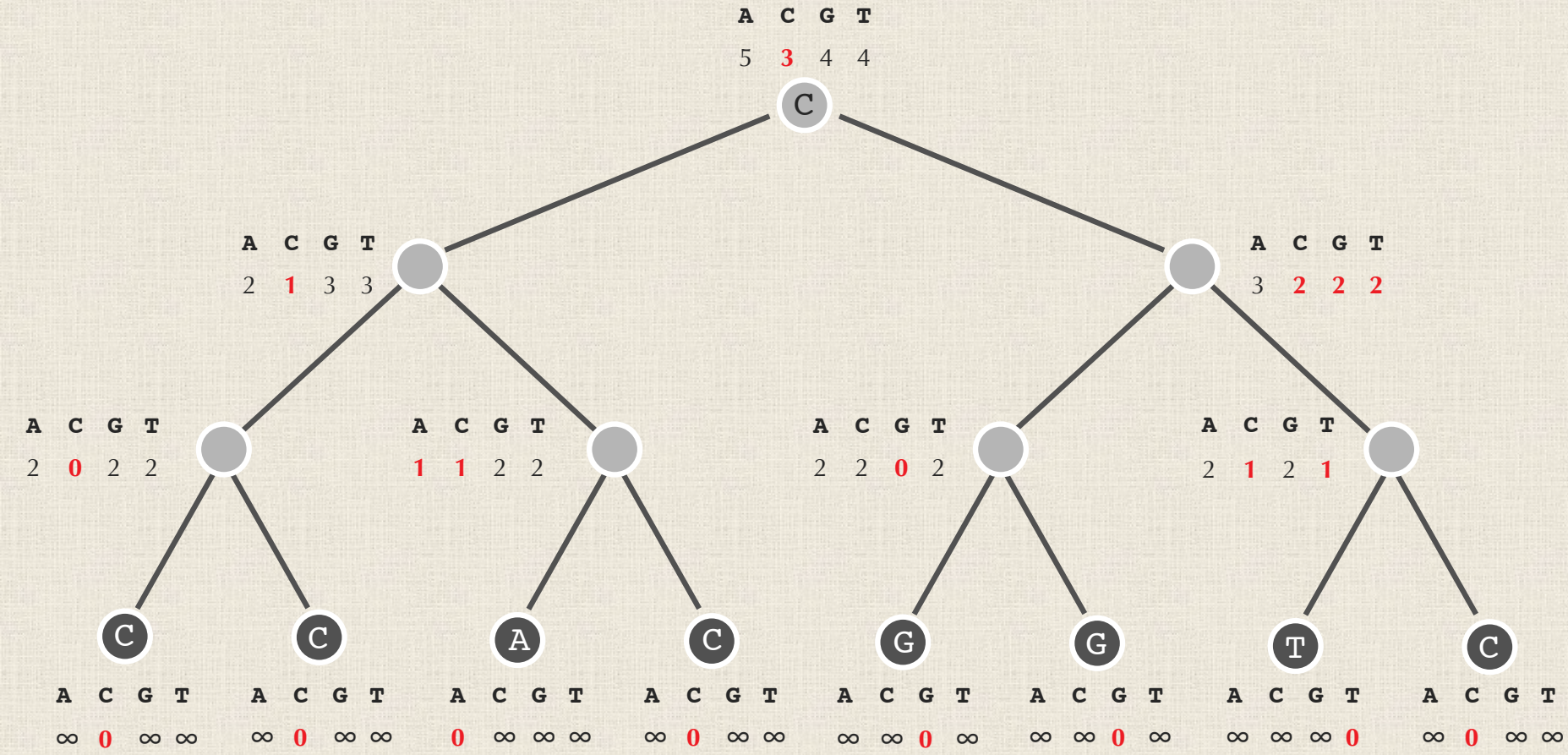

$$s_k(v) = \min_{\text{all symbols } i} \{s_i(Daughter(v)) + \alpha_{i,k}\} + \min_{\text{all symbols } i} \{s_i(Son(v)) + \alpha_{i,k}\}$$

# A Dynamic Programming Algorithm



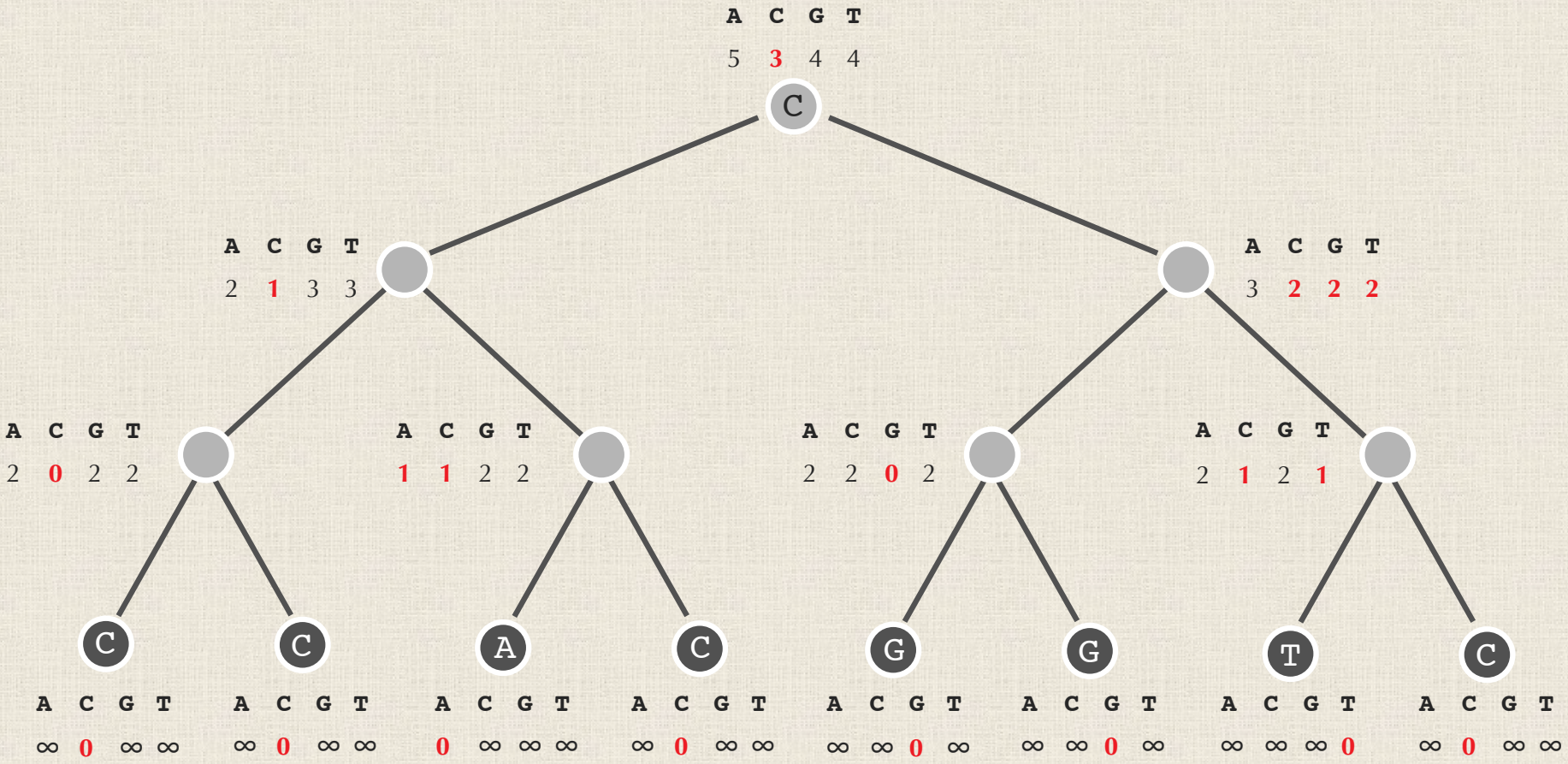

$$s_k(v) = \min_{\text{all symbols } i} \{s_i(Daughter(v)) + \alpha_{i,k}\} + \min_{\text{all symbols } i} \{s_i(Son(v)) + \alpha_{i,k}\}$$

# A Dynamic Programming Algorithm



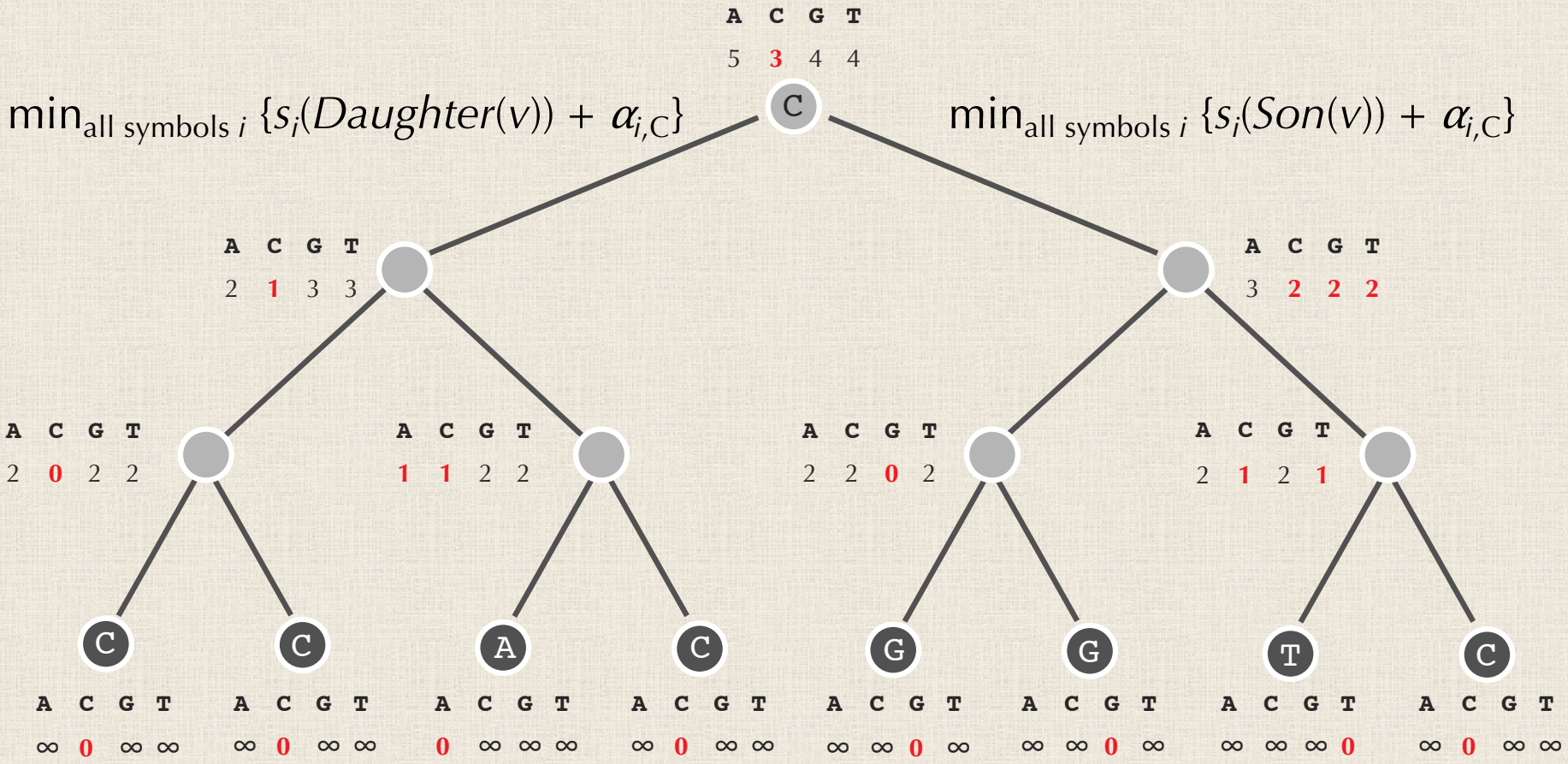

Because $s_k(root)$ is minimized when $k$ is C, we infer that the root must be assigned C!

# A Dynamic Programming Algorithm



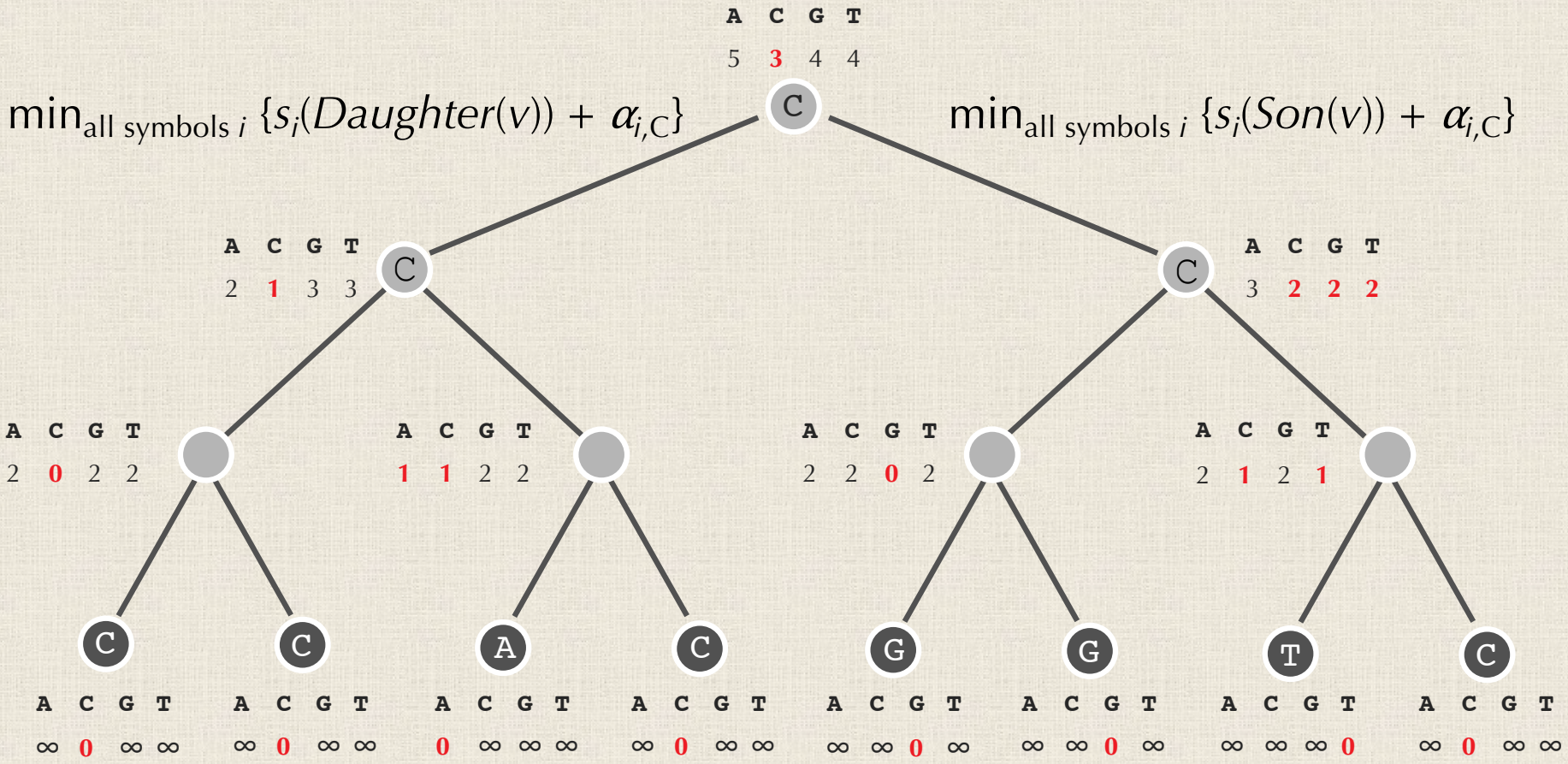

**STOP:** How should we "backtrack" to fill in the remaining nodes of the tree?

# A Dynamic Programming Algorithm



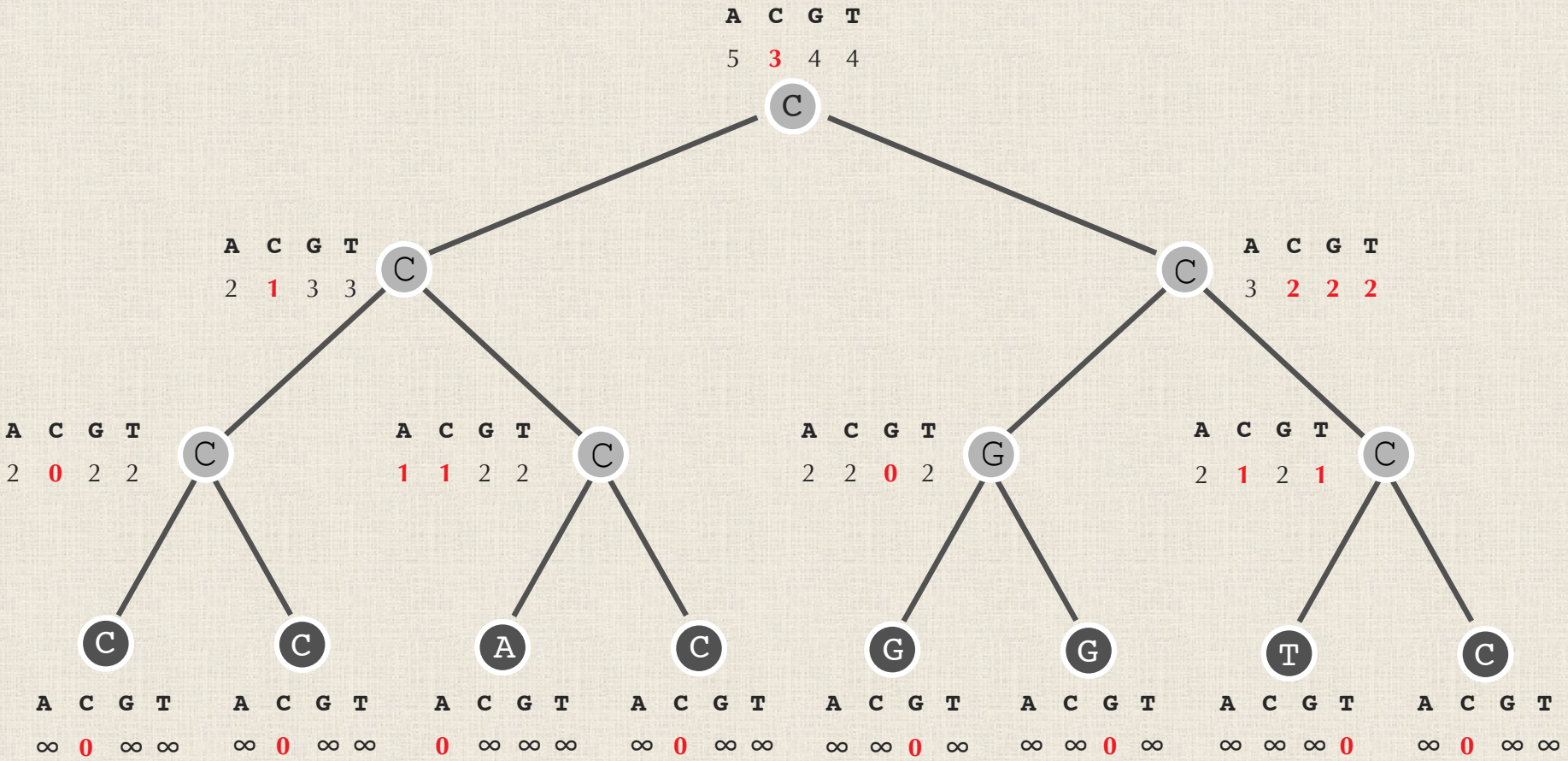

**Answer:** Remember which symbol was used at each child when computing $s_k(v)$!

# A Dynamic Programming Algorithm




**STOP:** Fill in the remaining nodes.

# A Dynamic Programming Algorithm



# Citations for Neighbor-Joining and Small Parsimony

**The neighbor-joining method: a new method for reconstructing ...**

by N Saitou · 1987 · Cited by 61260 · Related articles

The principle of this **method** is to find pairs of operational taxonomic units (OTUs [= neighbors]) that minimize the total branch length at each stage of clustering of OTUs starting with a starlike **tree**. The branch lengths as well as the topology of a parsimonious **tree** can quickly be obtained by using this **method**.

**A note on the neighbor-joining algorithm of Saitou and Nei**

by JA Studier · 1988 · Cited by 614 · Related articles

**A note on the neighbor-joining algorithm of Saitou and Nei**. Mol Biol Evol. 1988 Nov;5(6):729-31. doi: 10.1093/oxfordjournals.molbev.a040527.

**Minimal Mutation Trees of Sequences - JSTOR**

by D Sankoff · 1975 · Cited by 649 · Related articles

The integer d(x, y) equals the **minimum** number of **mutations** required to transform **sequence** x into y, or y into x, where a **mutation** may be either a change (replacement) of the value in A of a single term x(i) to correspond with the value of some y(j), or else the deletion from, or insertion into **sequence** x, of a single ...