Genome 570, Phylogenetic Inference

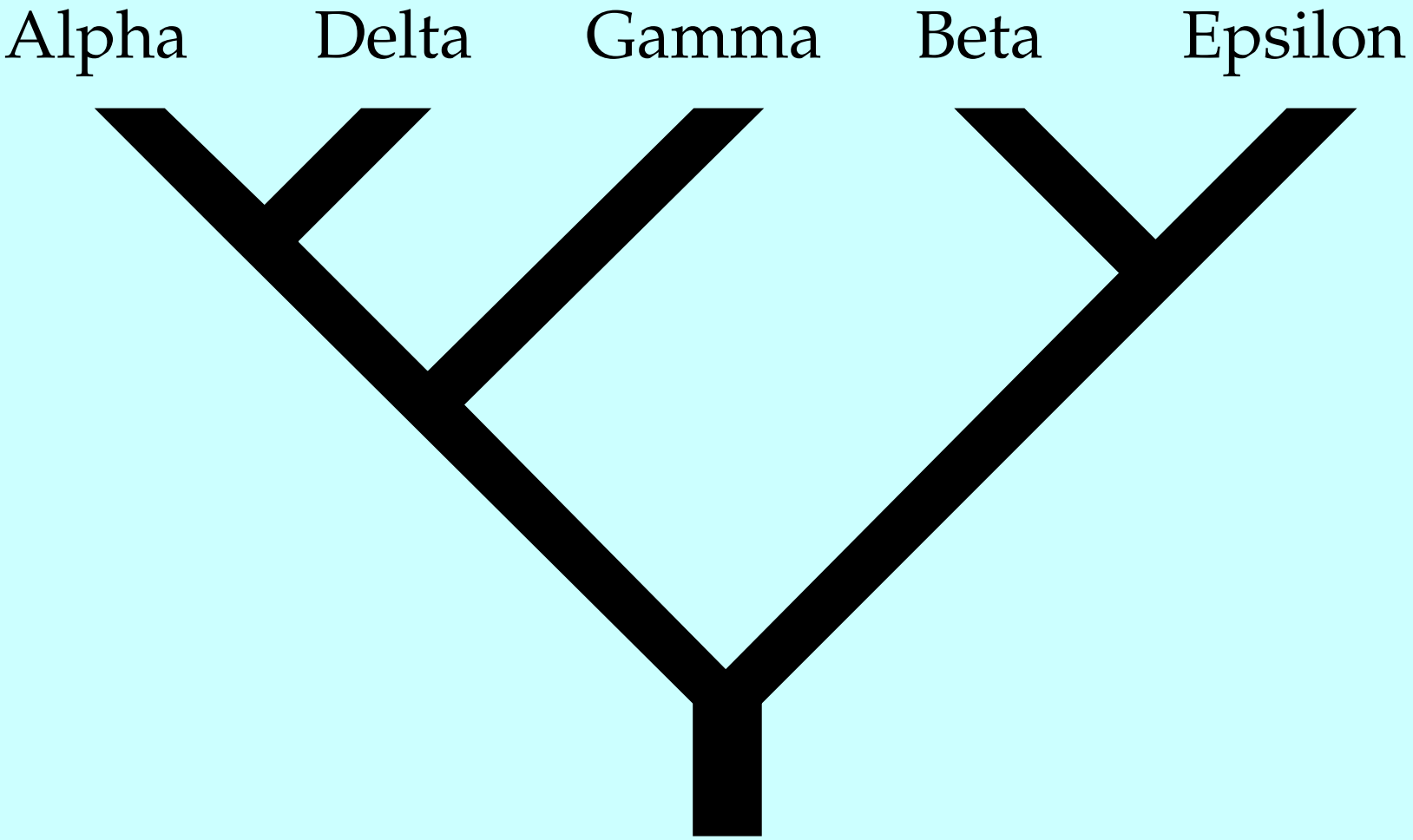
January 2016

Week 1: Parsimony, tree enumeration

A simple data set

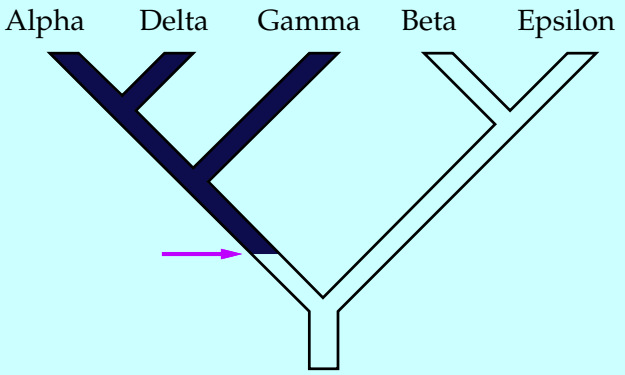
Species	Characters					
	1	2	3	4	5	6
Alpha	1	0	0	1	1	0
Beta	0	0	1	0	0	0
Gamma	1	1	0	0	0	0
Delta	1	1	0	1	1	1
Epsilon	0	0	1	1	1	0

The tree we will evaluate

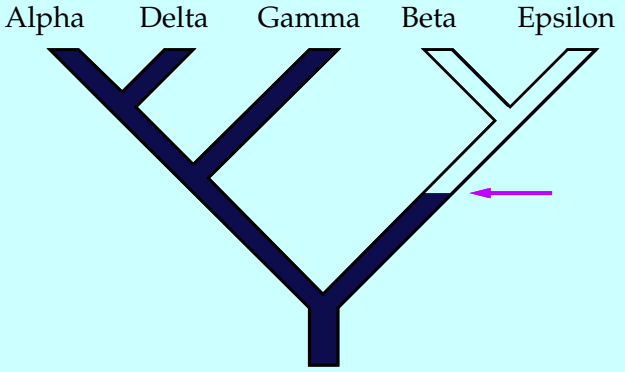


Character 1

	1	2	3	4	5	6
Alpha	1	0	0	1	1	0
Beta	0	0	1	0	0	0
Gamma	1	1	0	0	0	0
Delta	1	1	0	1	1	1
Epsilon	0	0	1	1	1	0

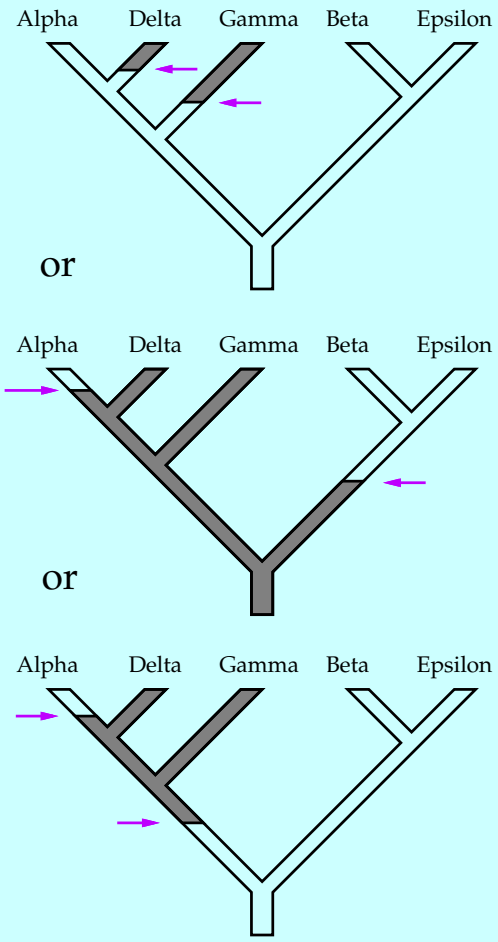


or

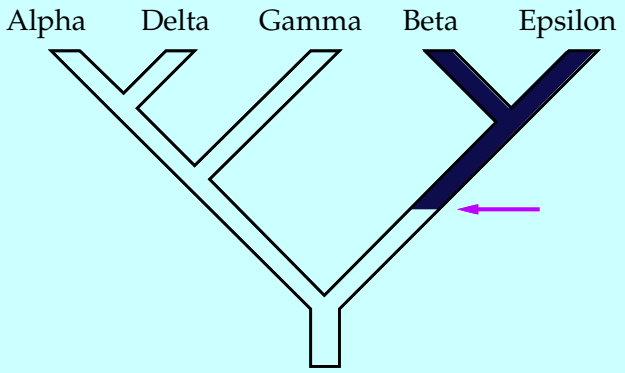


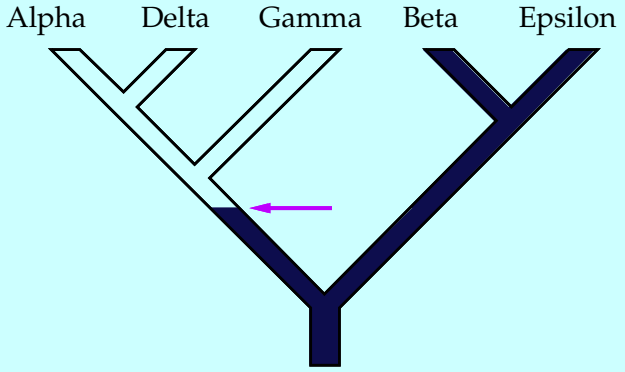
Character 2

	1	2	3	4	5	6
Alpha	1	0	0	1	1	0
Beta	0	0	1	0	0	0
Gamma	1	1	0	0	0	0
Delta	1	1	0	1	1	1
Epsilon	0	0	1	1	1	0



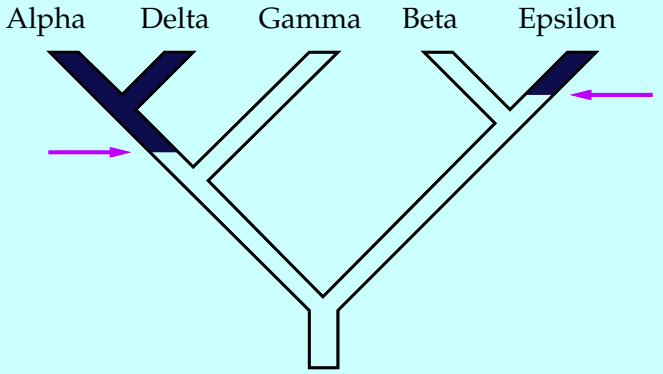
Character 3



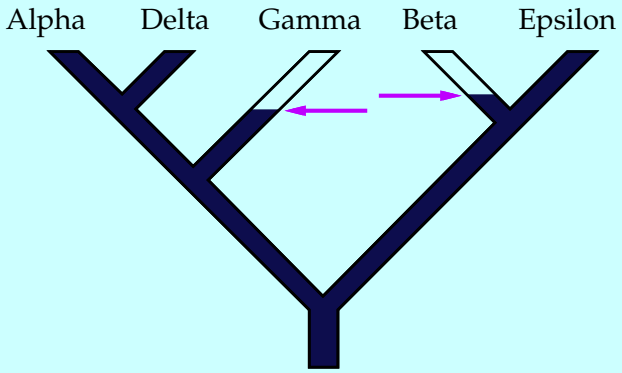


	1	2	3	4	5	6
Alpha	1	0	0	1	1	0
Beta	0	0	1	0	0	0
Gamma	1	1	0	0	0	0
Delta	1	1	0	1	1	1
Epsilon	0	0	1	1	1	0

Character 4 (and 5)

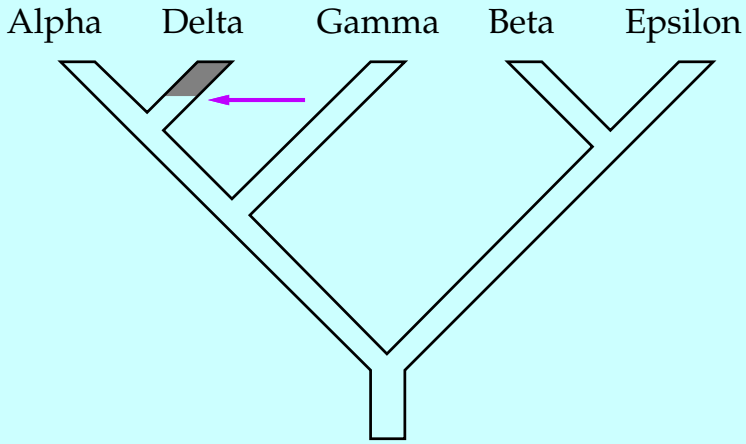


or



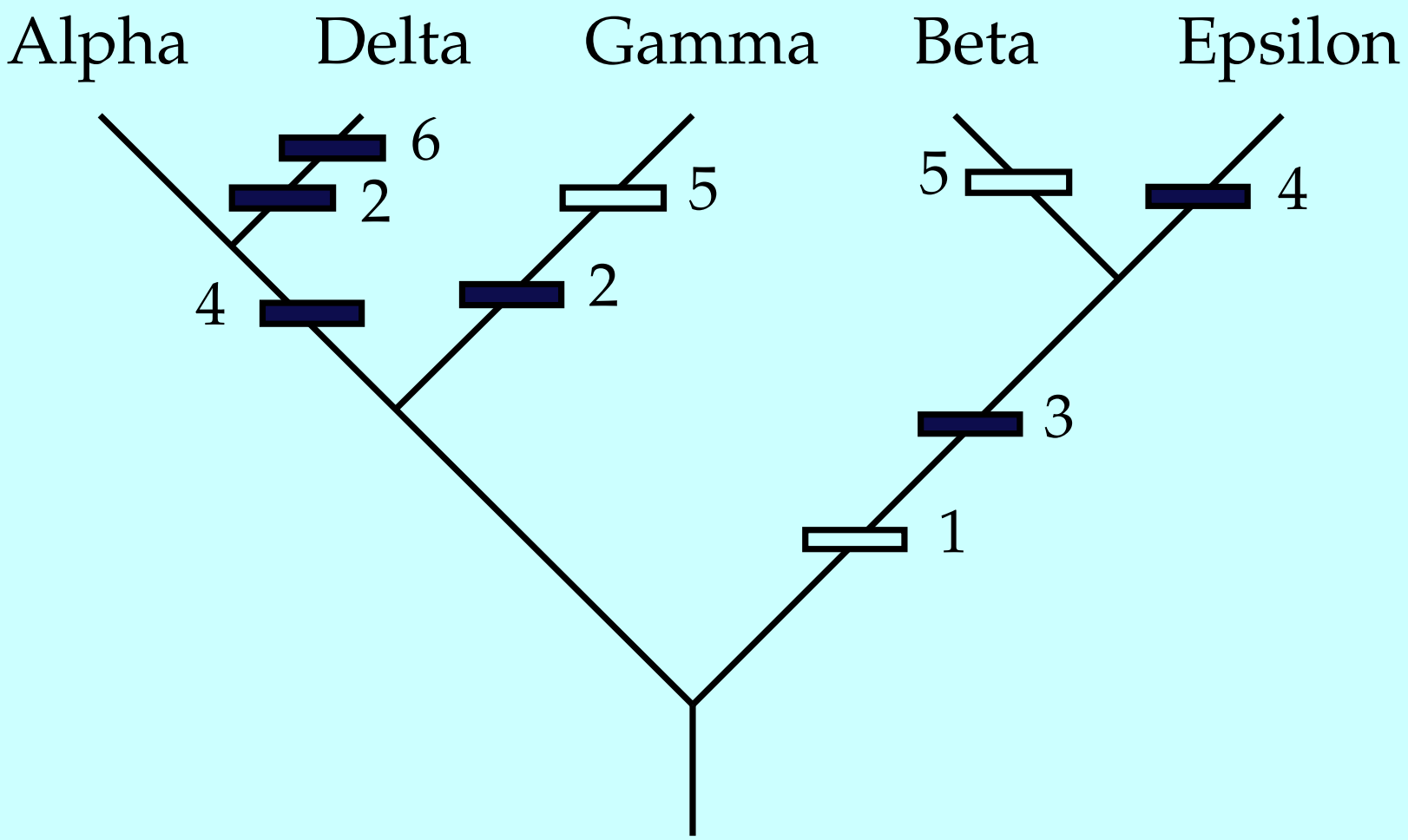
	1	2	3	4	5	6
Alpha	1	0	0	1	1	0
Beta	0	0	1	0	0	0
Gamma	1	1	0	0	0	0
Delta	1	1	0	1	1	1
Epsilon	0	0	1	1	1	0

Character 6



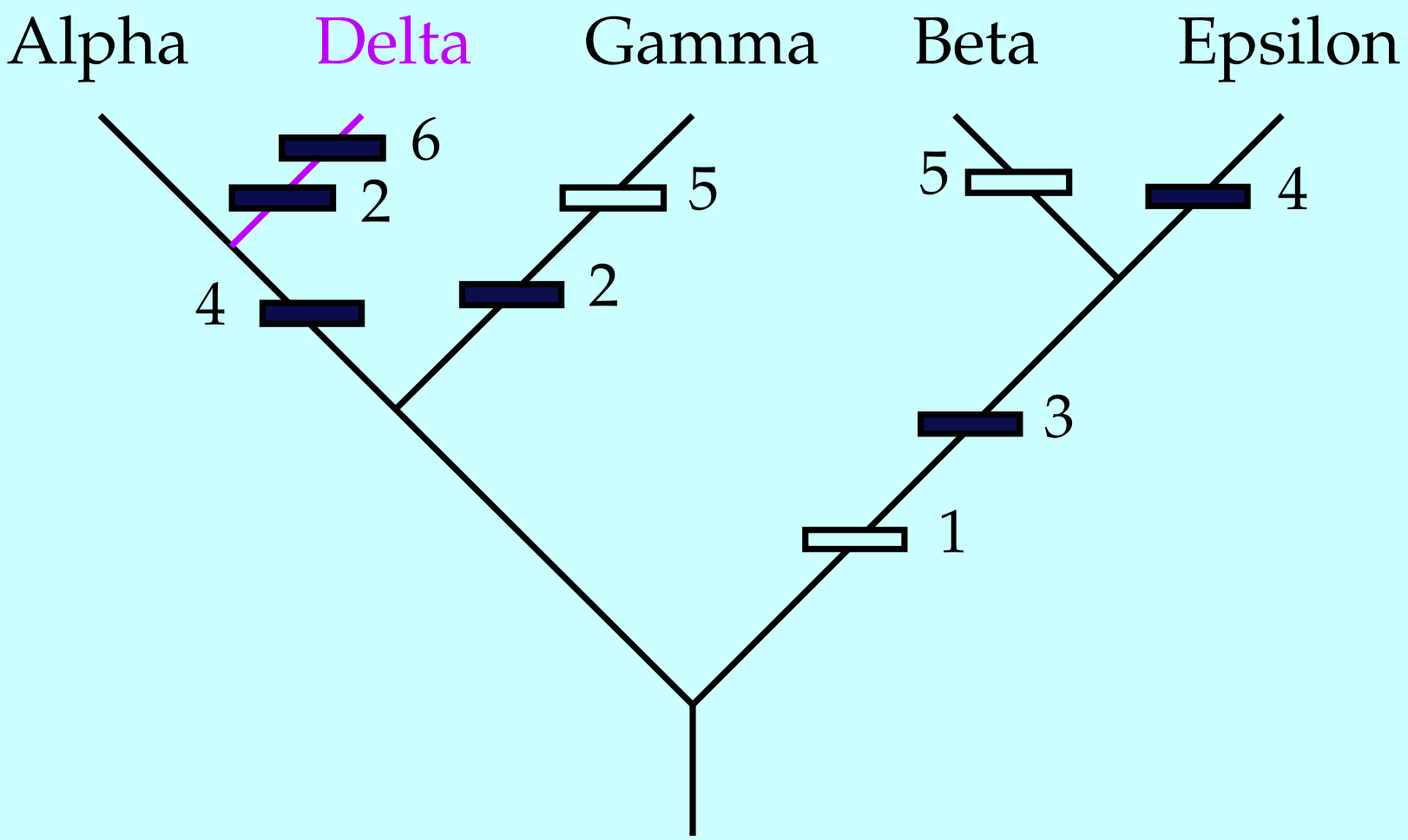
	1	2	3	4	5	6
Alpha	1	0	0	1	1	0
Beta	0	0	1	0	0	0
Gamma	1	1	0	0	0	0
Delta	1	1	0	1	1	1
Epsilon	0	0	1	1	1	0

Changes in all characters



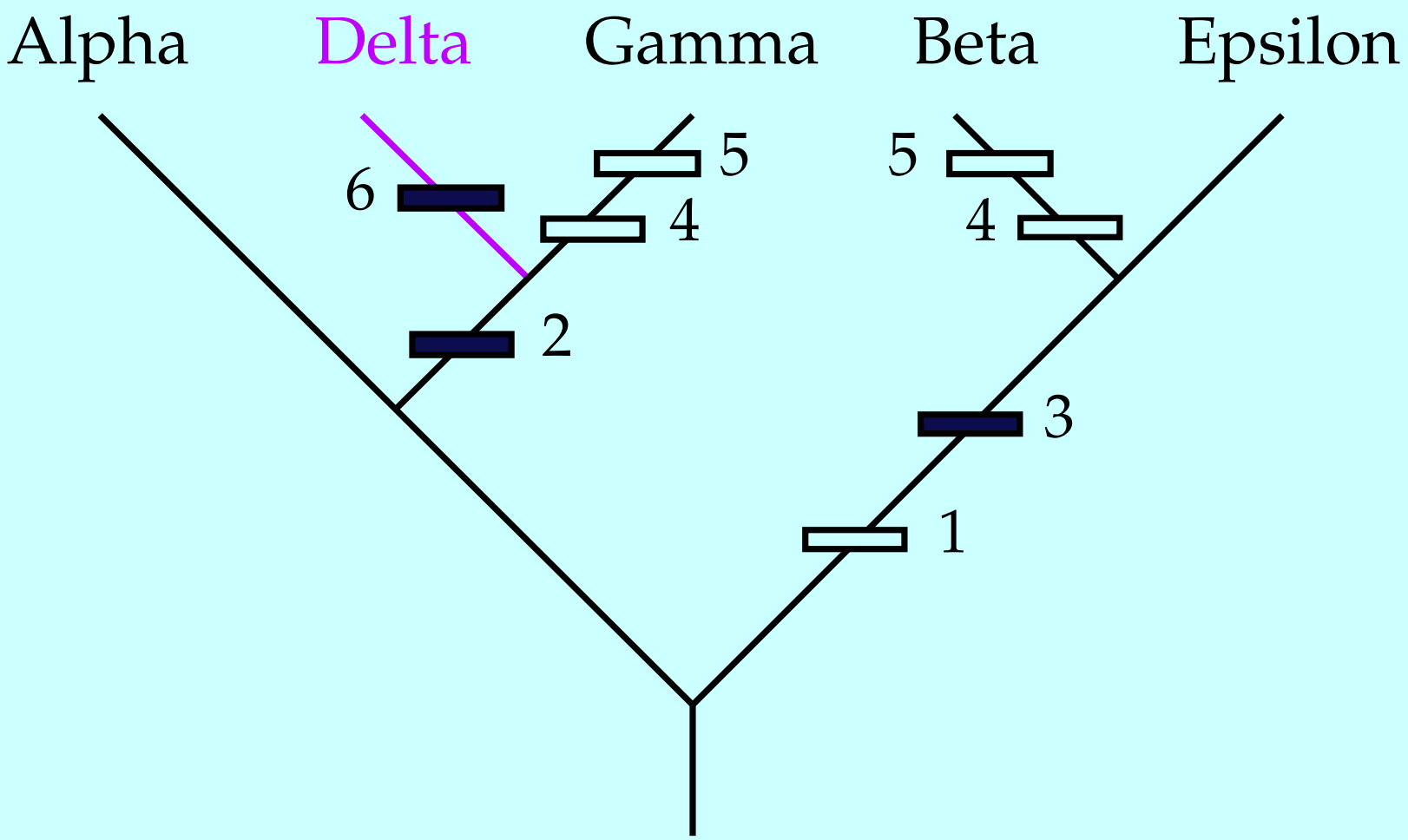
There are a total of 9 changes.

Finding a better tree



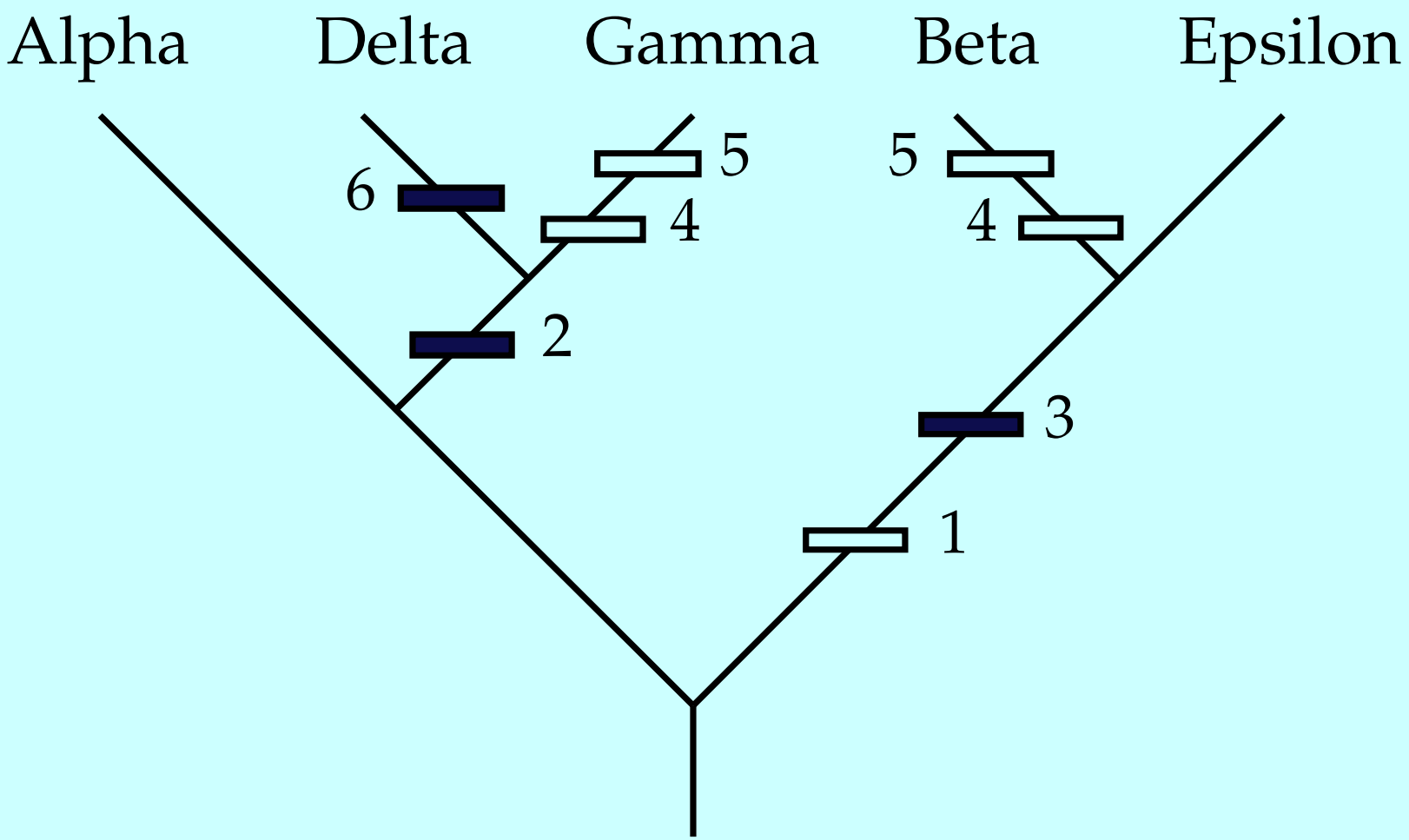
Moving one species, Delta ...

Moving it to a new position ...



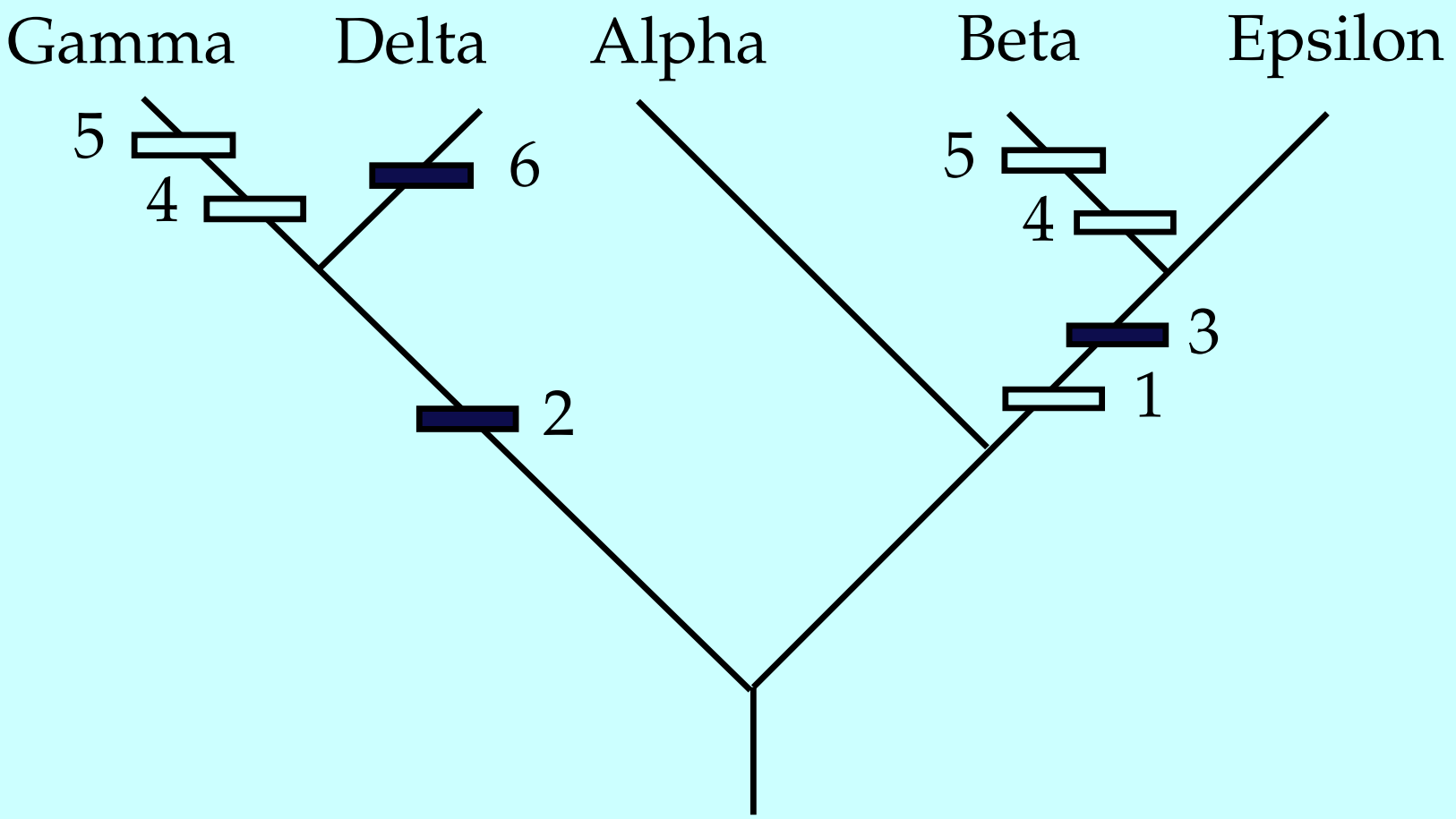
We end up with a tree with 8 changes.

Changes on the most parsimonious tree

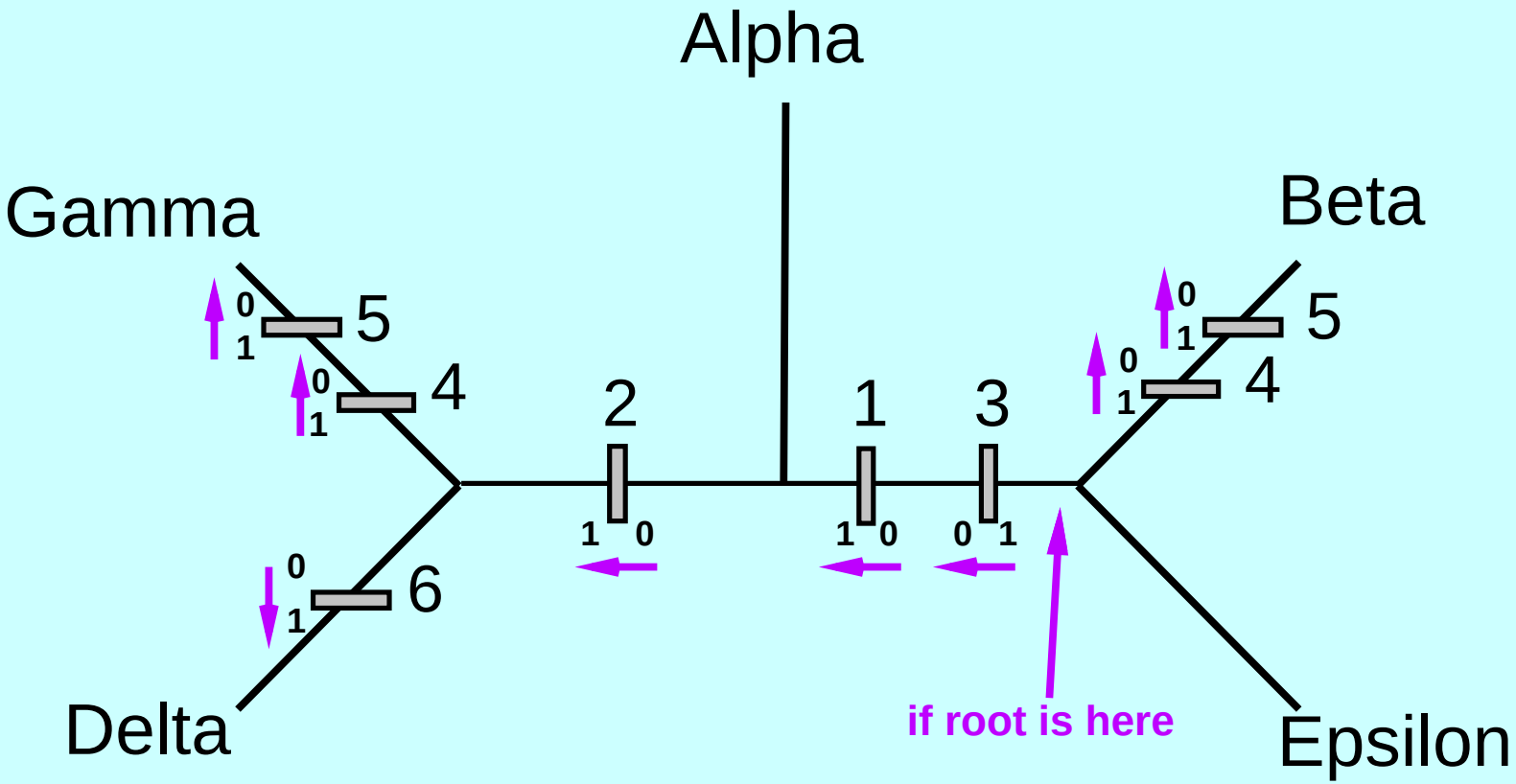


And 8 turns out to be the smallest possible number of changes.

Another rooted tree with same number of changes

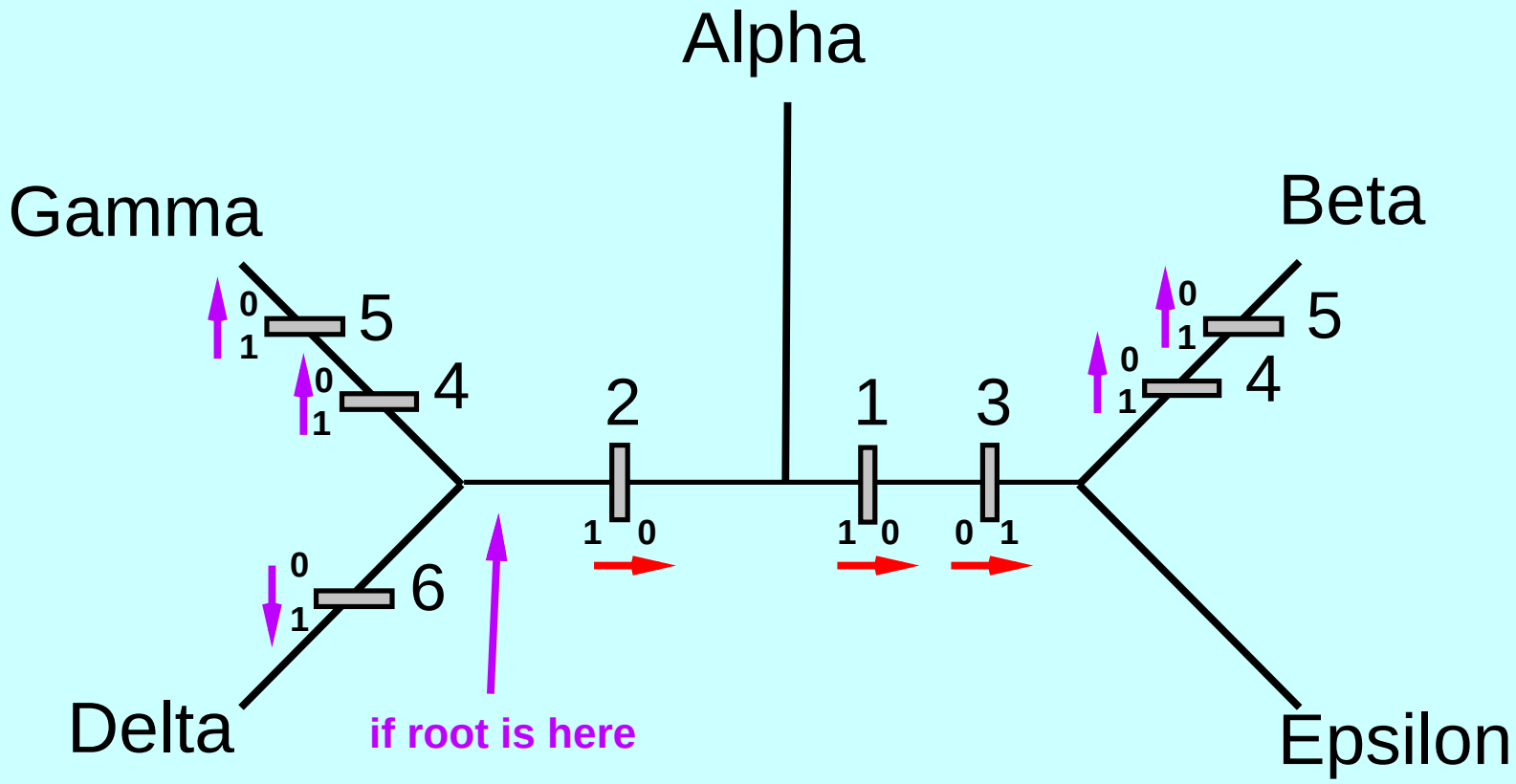


The corresponding unrooted tree



The changes placed on the tree really delimit regions of the tree that are reconstructed to have particular states.

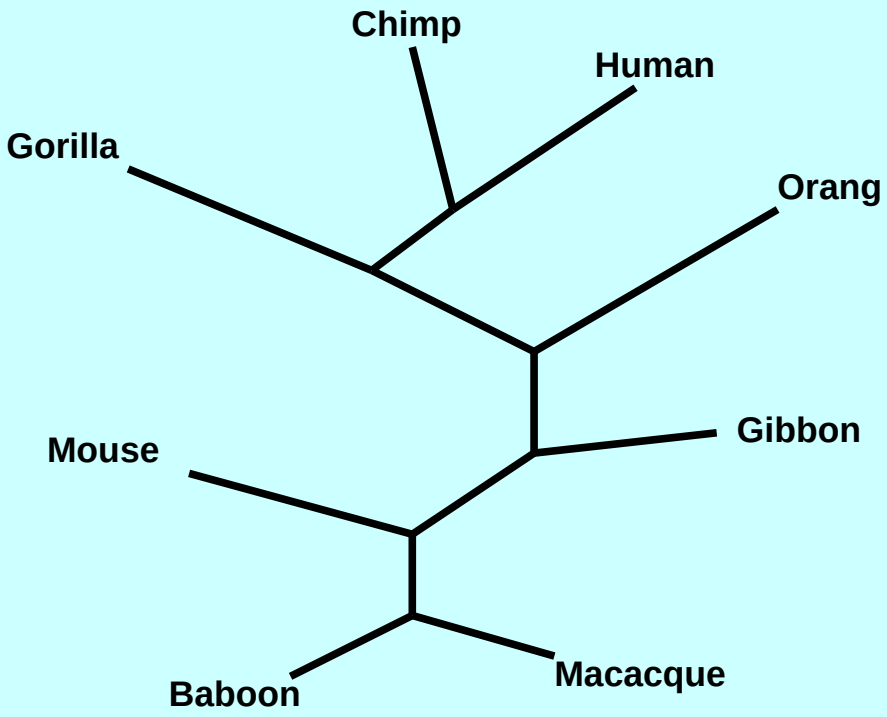
The corresponding unrooted tree



If we place the root of the tree in a different location, the direction of some of the changes is altered, but not the number of these boundaries of the state regions.

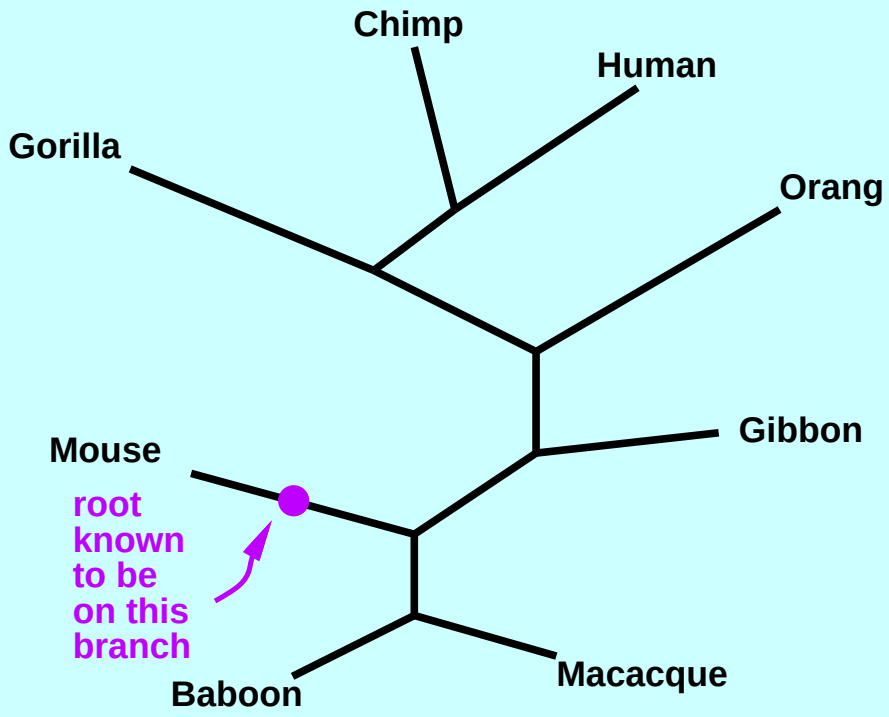
(This is true if changes in both directions have the same “cost”).

Using an outgroup to root an unrooted tree



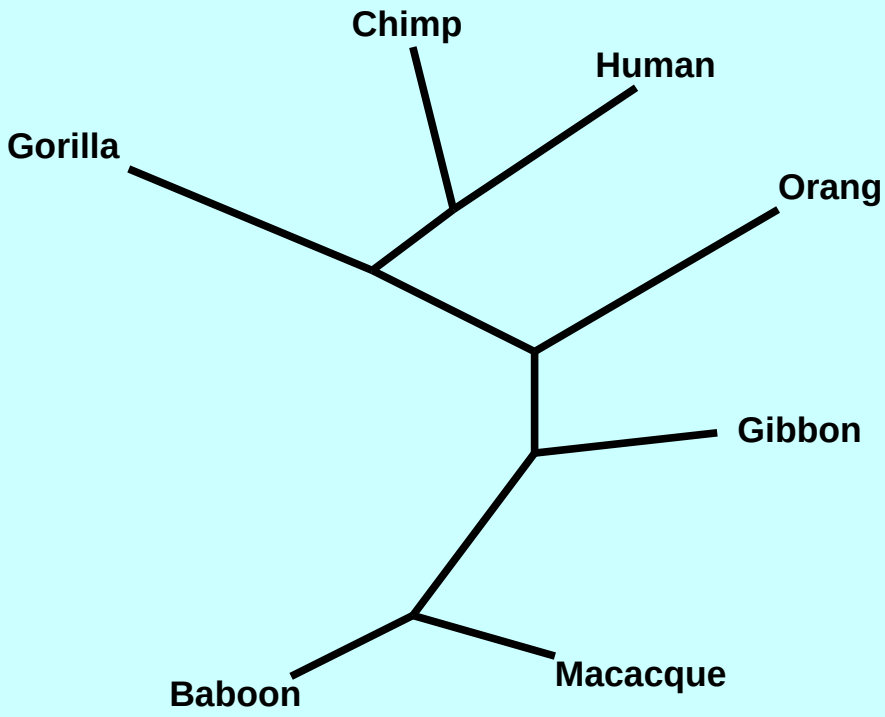
If we have inferred an unrooted tree ...

Using an outgroup to root an unrooted tree



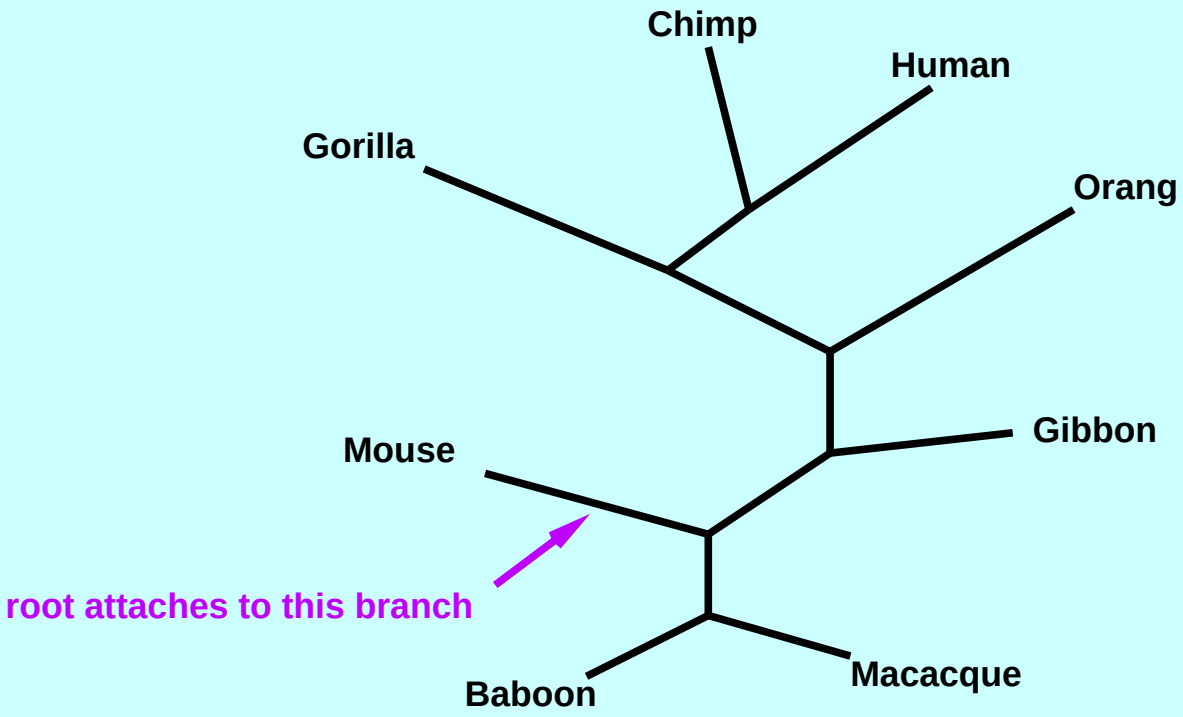
... and we know that one branch (here, the branch to Mouse) is the one where the root connects, then that prior knowledge roots the tree. That group or species is the “outgroup”. This amounts to the prior knowledge that the rest of the tree is a monophyletic group (the “ingroup”).

Using an outgroup to root an unrooted tree



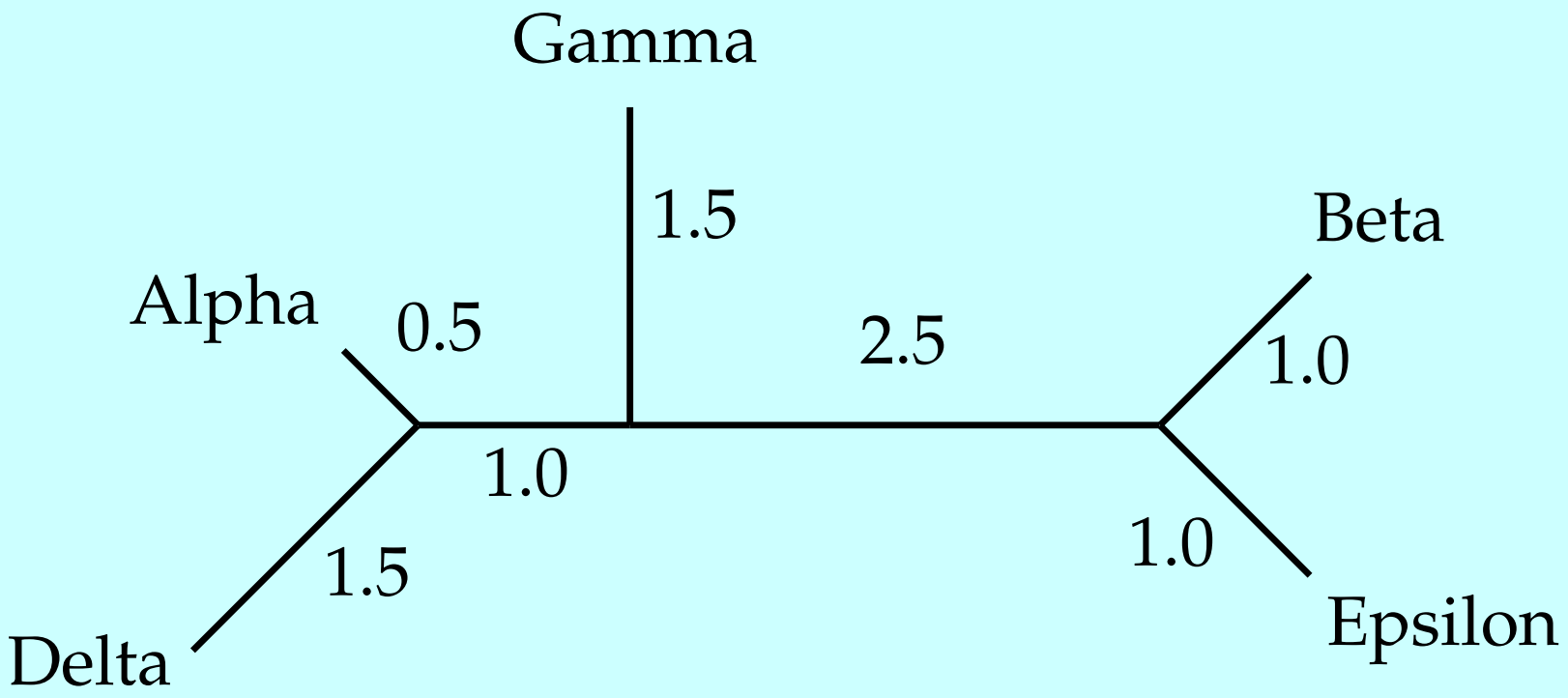
If instead we have this unrooted tree, and we add to the data set the species Mouse, which will be our outgroup ...

Using an outgroup to root an unrooted tree



... it adds information by attaching to one branch or another. Then we use our prior assumption that the root is on the branch between Mouse and whatever node to which it attaches.

Branch lengths, averaging over reconstructions



Walter Fitch (1929-2011)

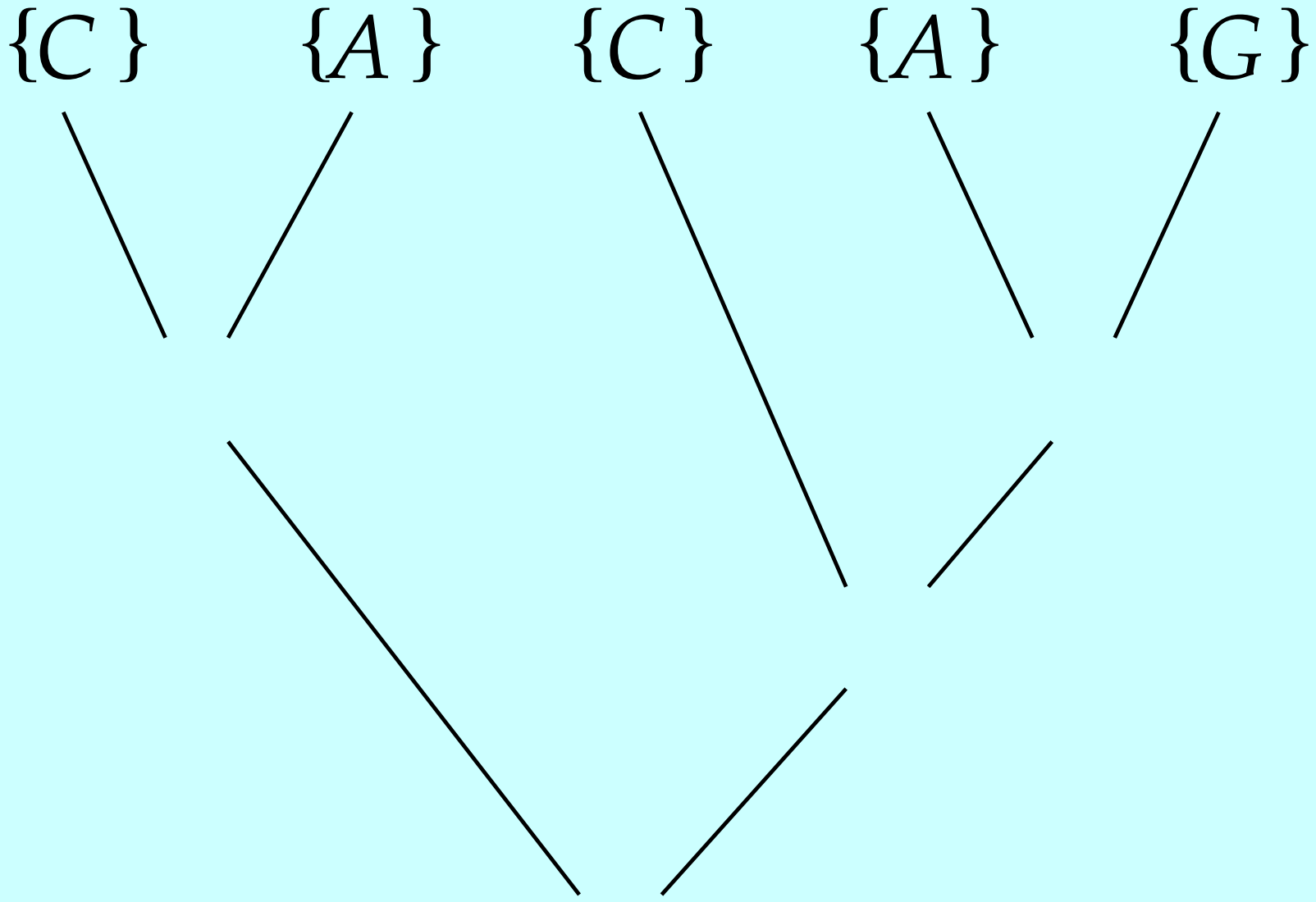
Walter Fitch, in 1975

The Fitch algorithm

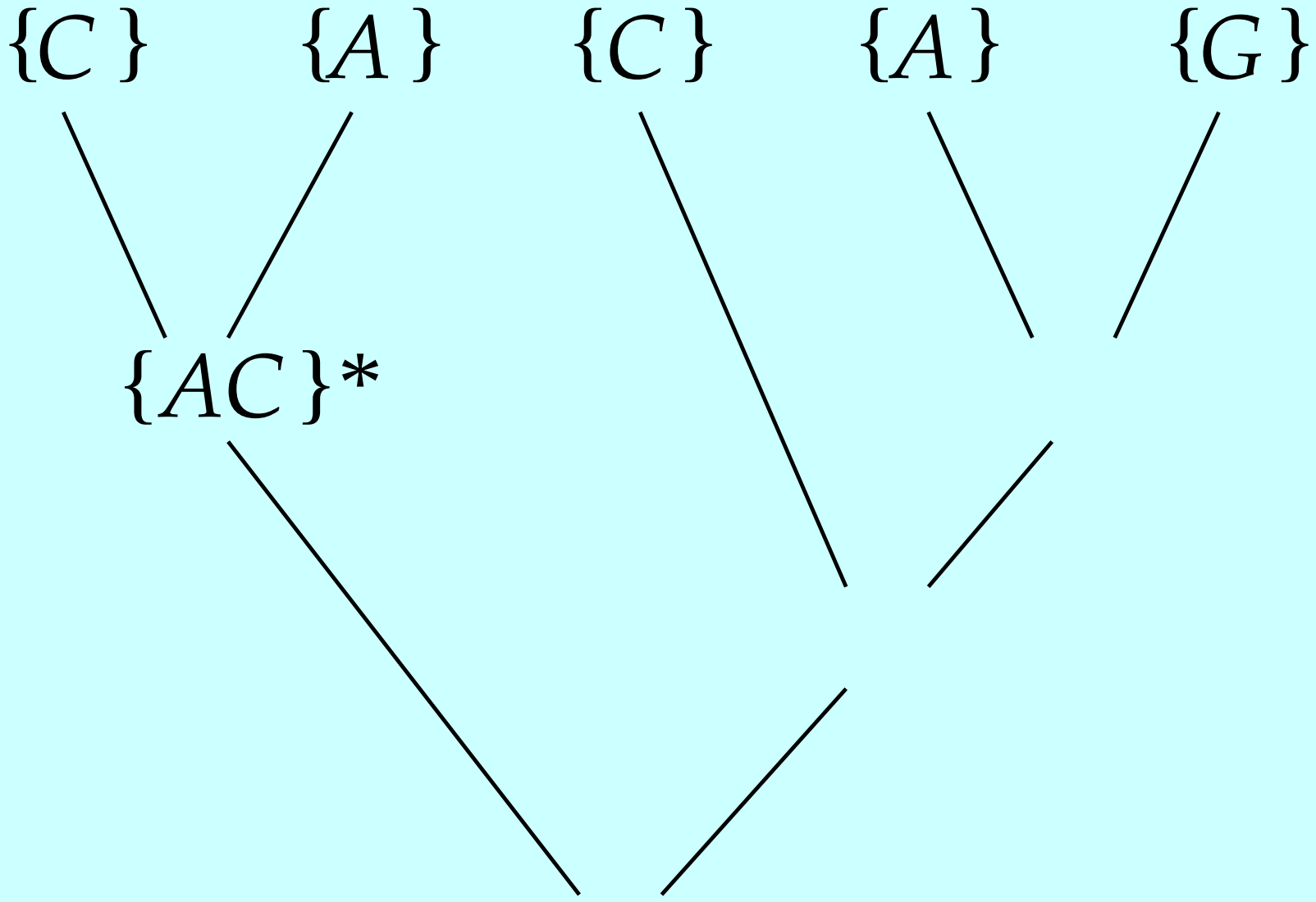
To count steps at a site:

1. At the tips of the tree, make a set with those nucleotides which are possible at that tip (more than one if there is ambiguity)
2. Go down the tree (perhaps by postorder tree traversal), considering each node only after its two descendants have been considered.
3. At each interior node, construct the set as the intersection of the two descendant sets:
$$S_n = S_\ell \cap S_r$$
4. If this is empty, instead construct the union of the two descendant sets, and count one step:
$$S_n = S_\ell \cup S_r$$
5. Continue to the bottom node. The count of steps is the total of these counts.

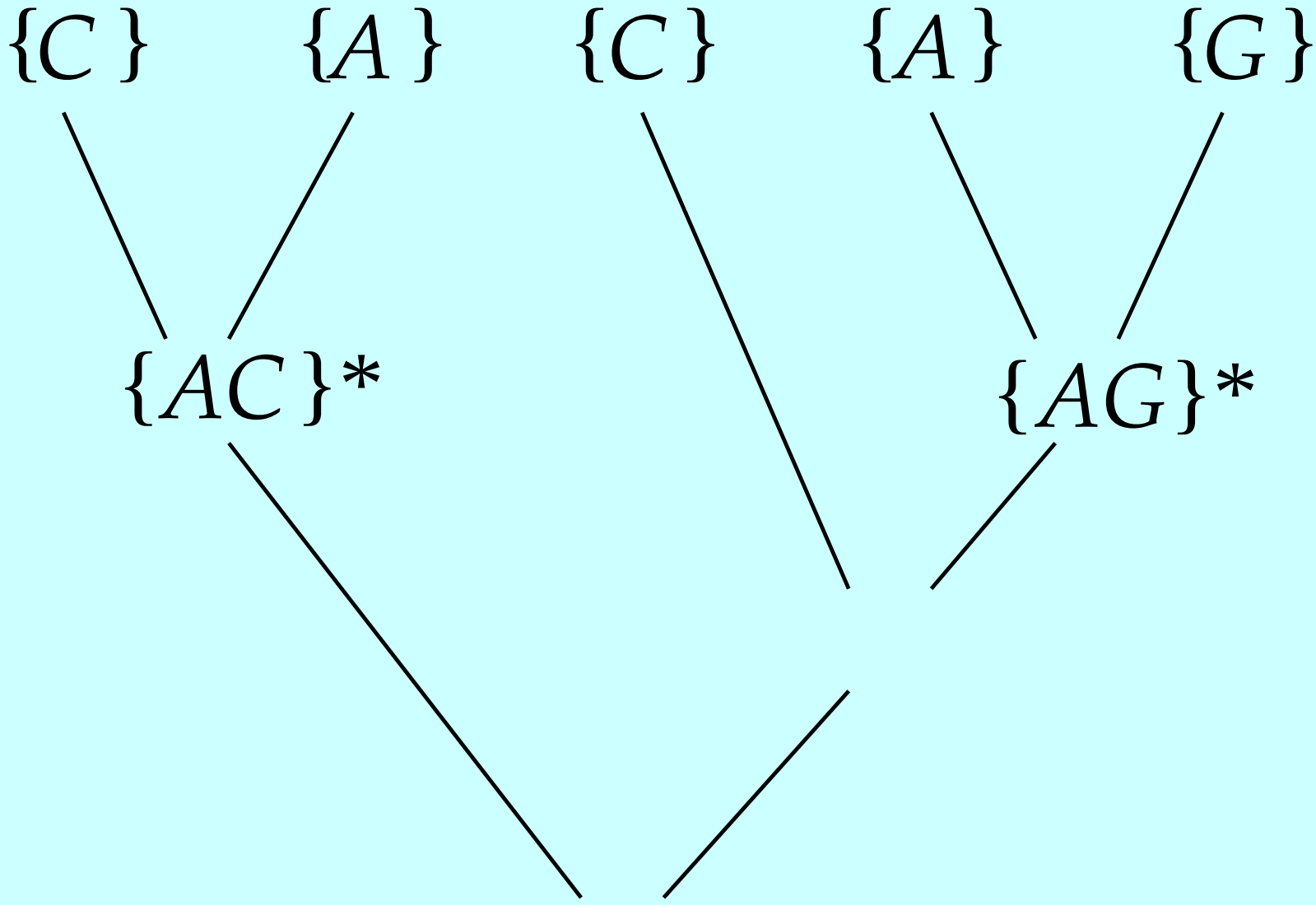
Example for the Fitch algorithm



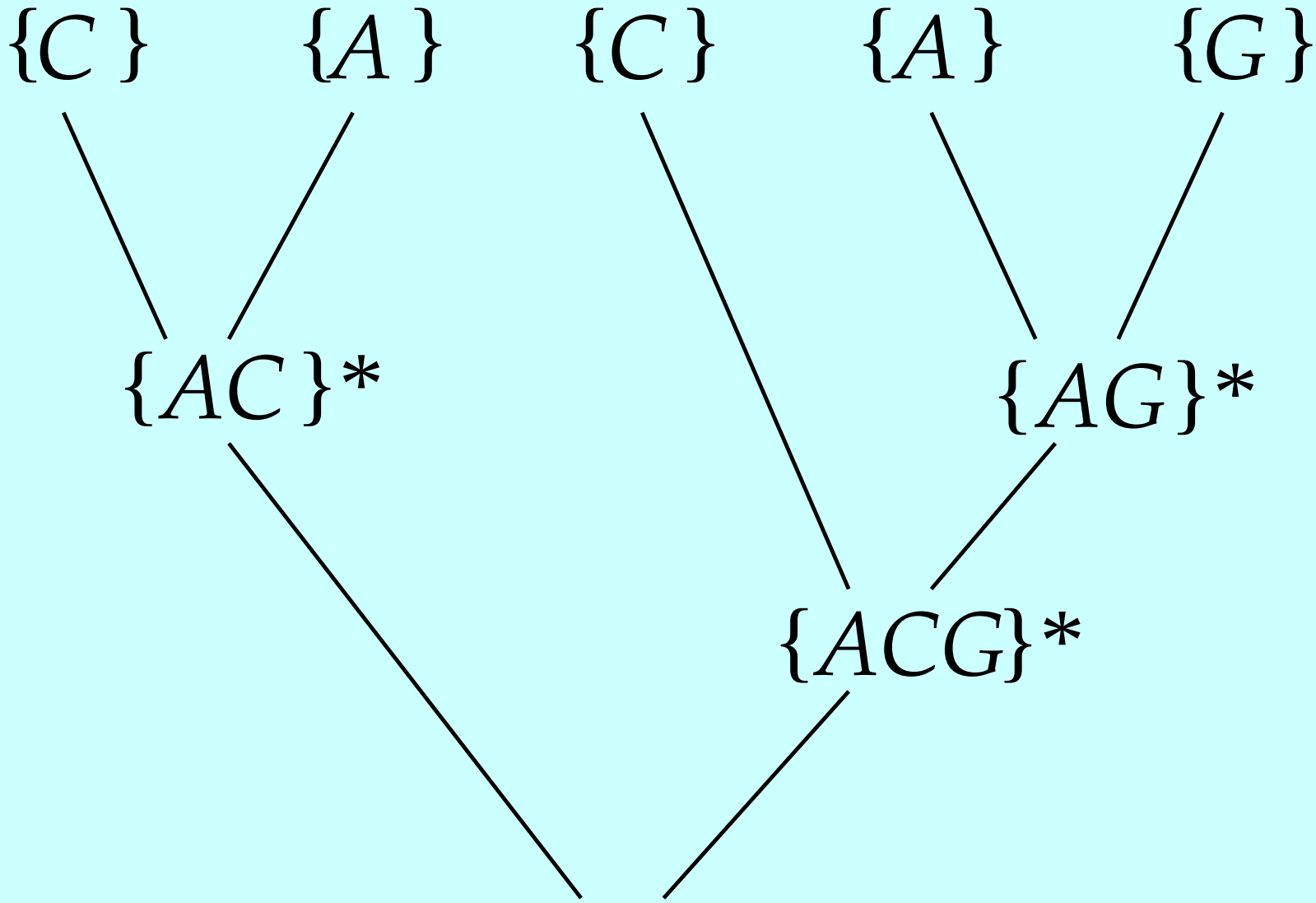
Example for the Fitch algorithm



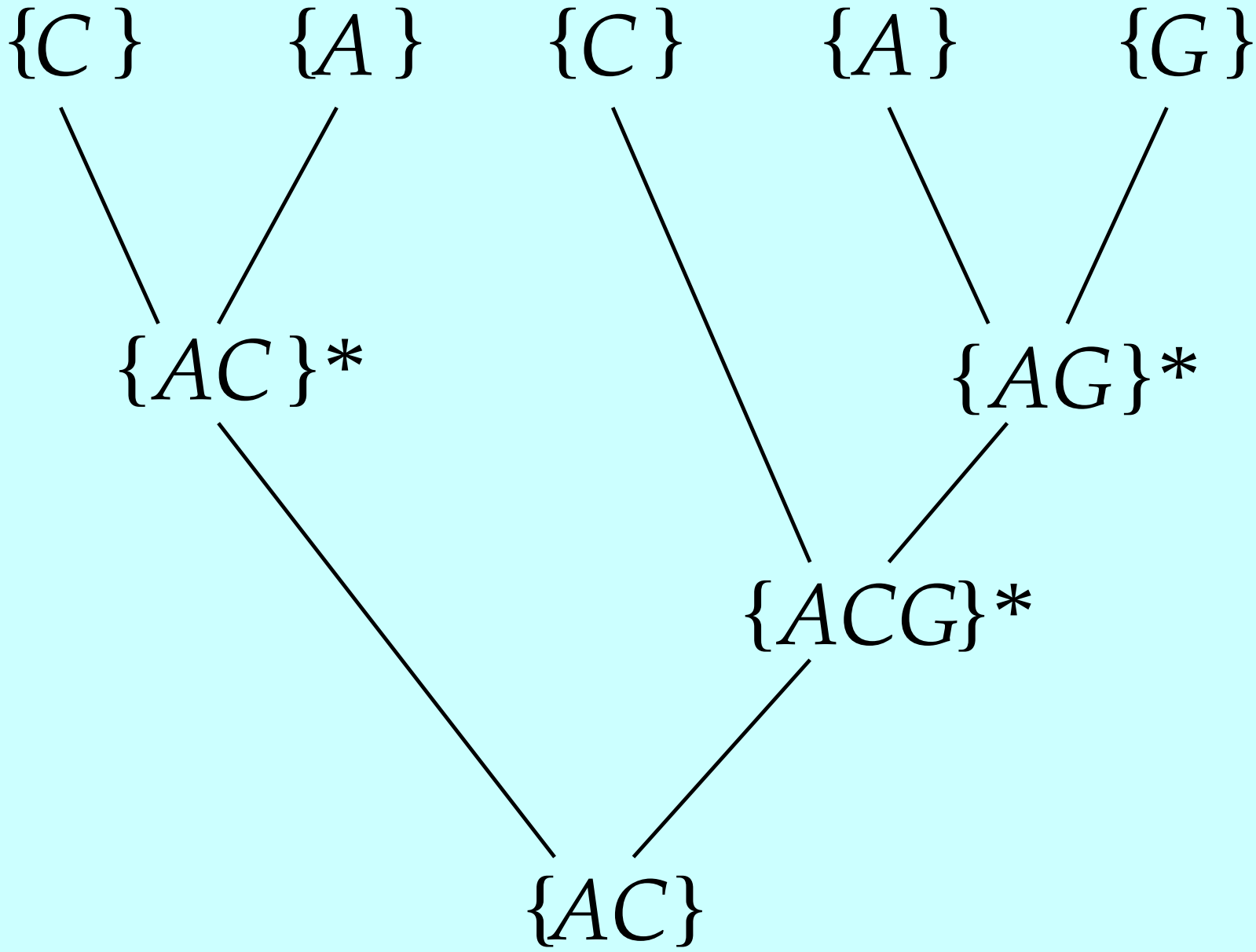
Example for the Fitch algorithm



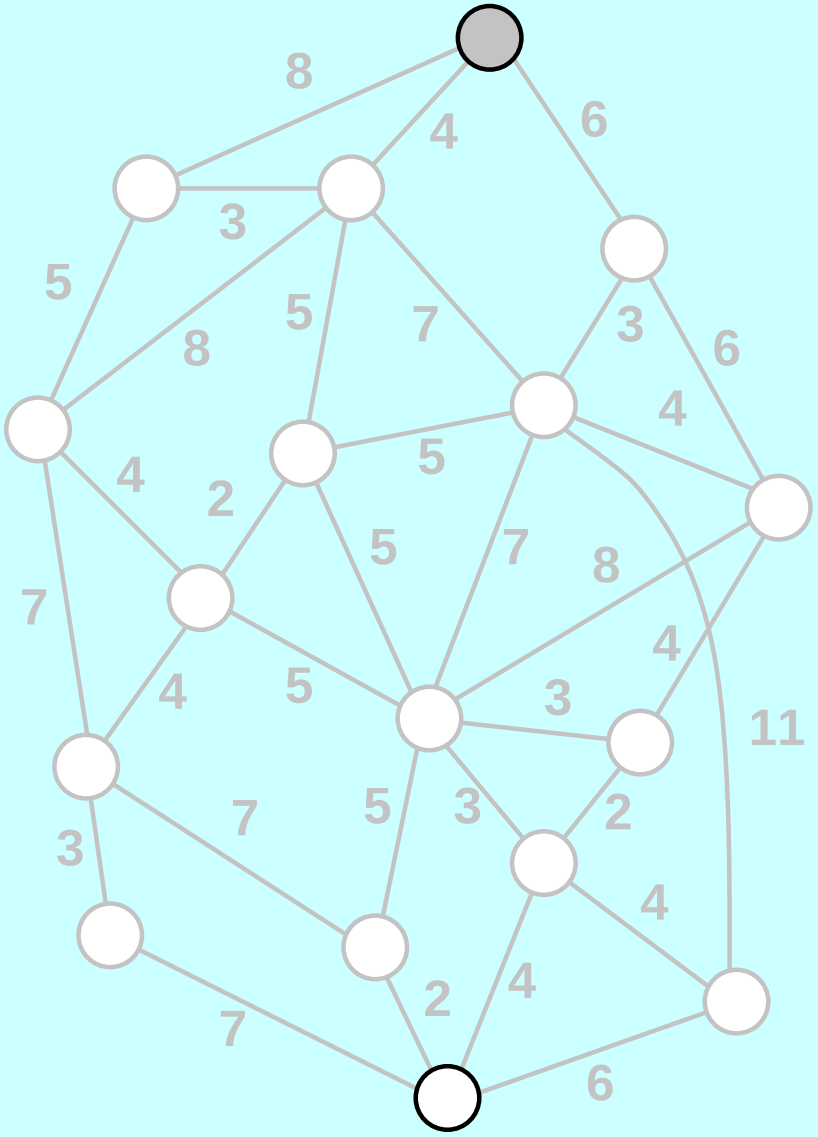
Example for the Fitch algorithm



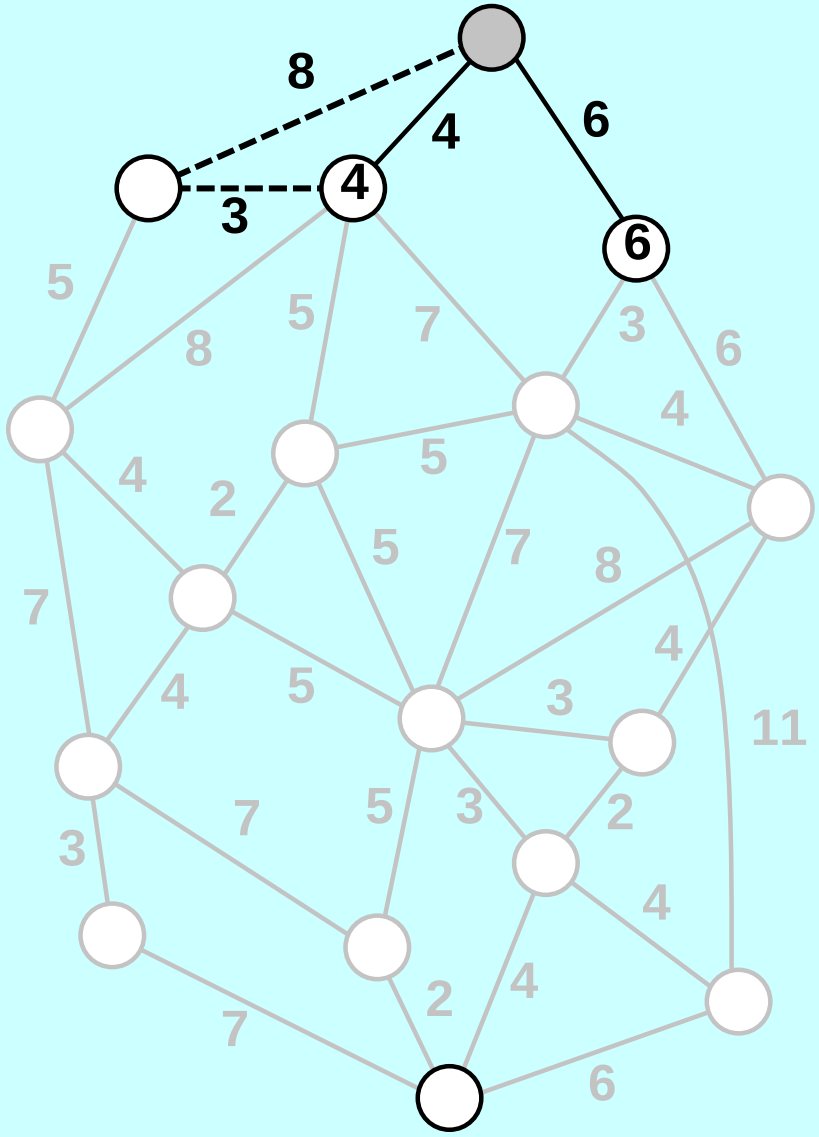
Example for the Fitch algorithm



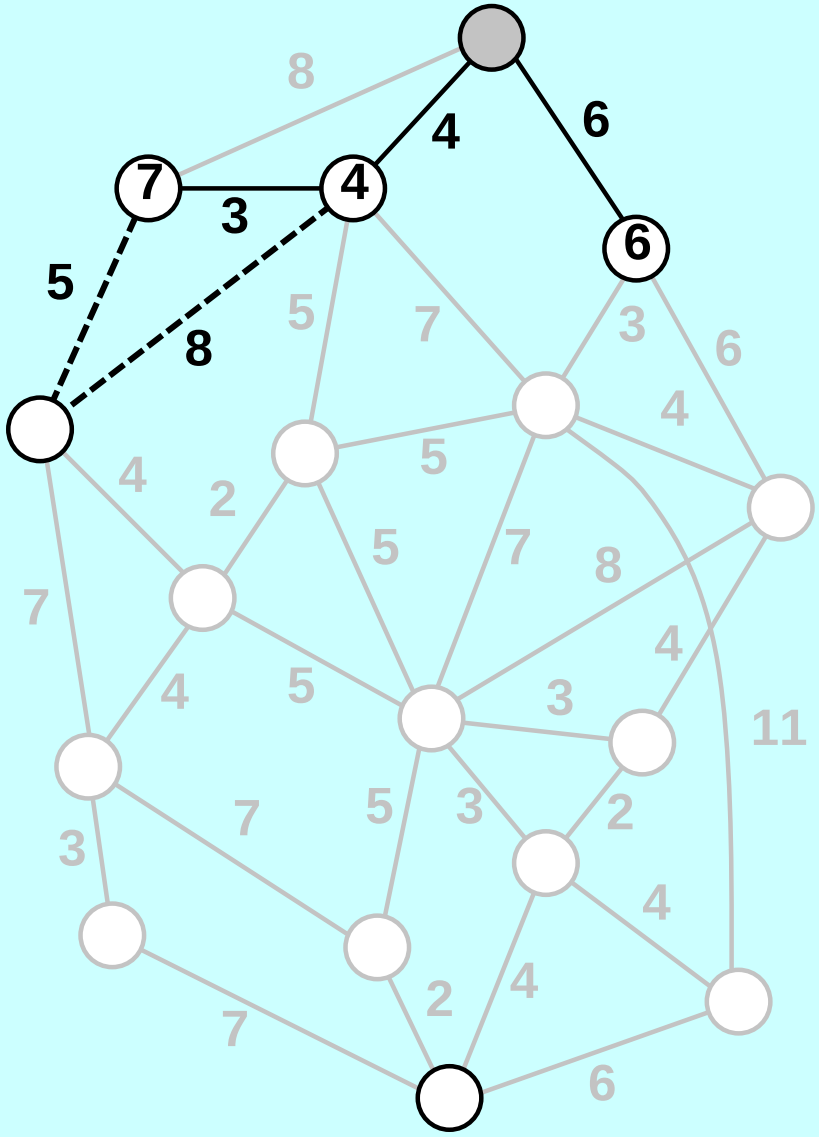
Shortest path through a graph



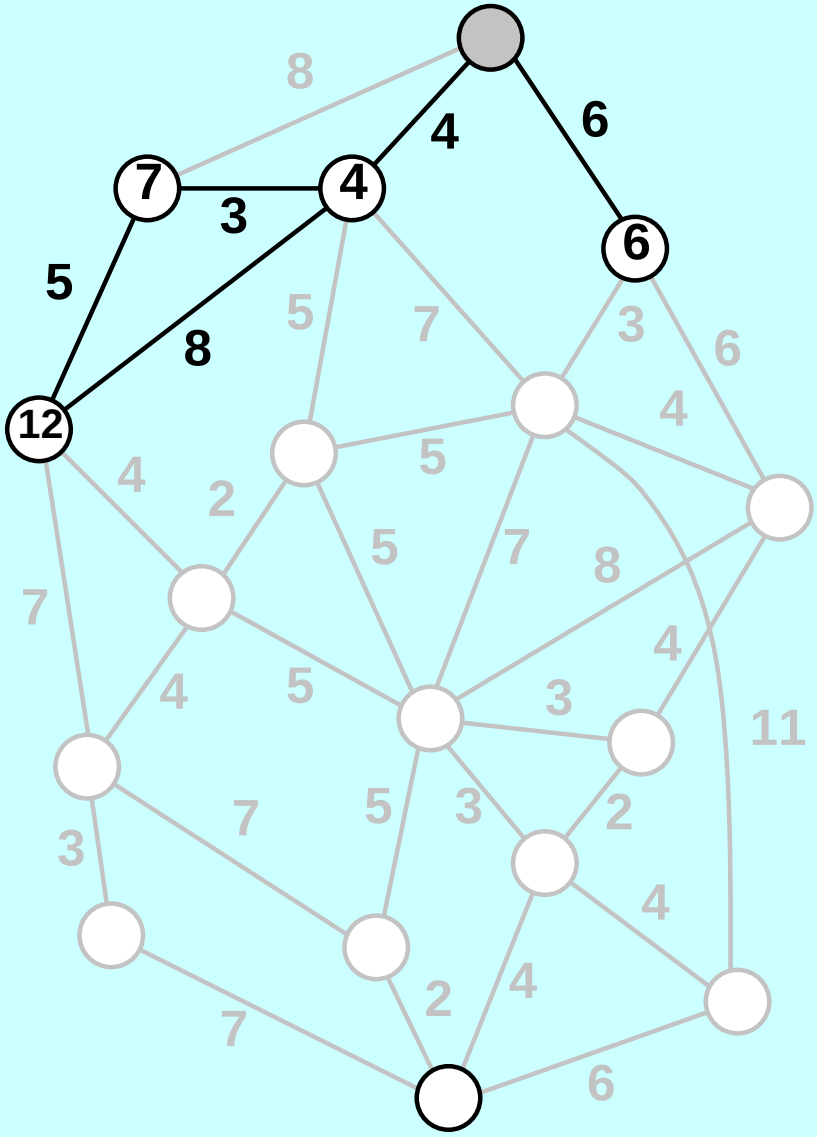
Shortest path through a graph



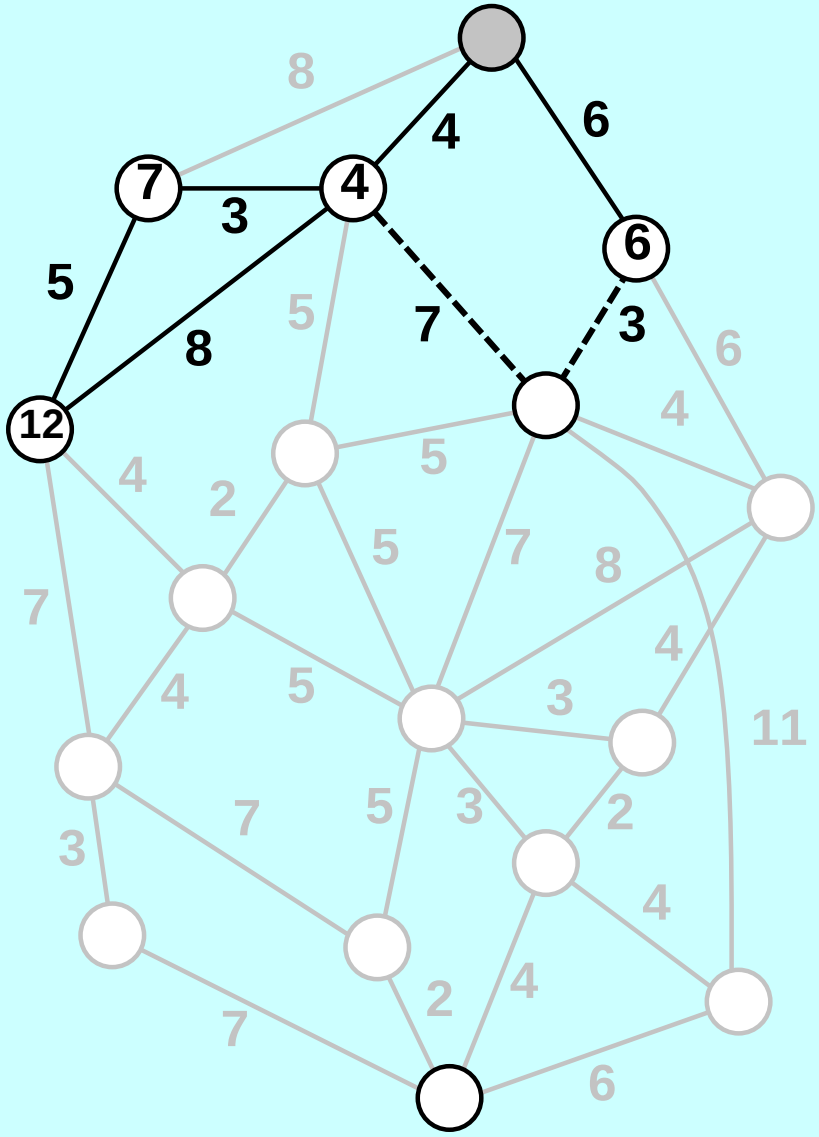
Shortest path through a graph



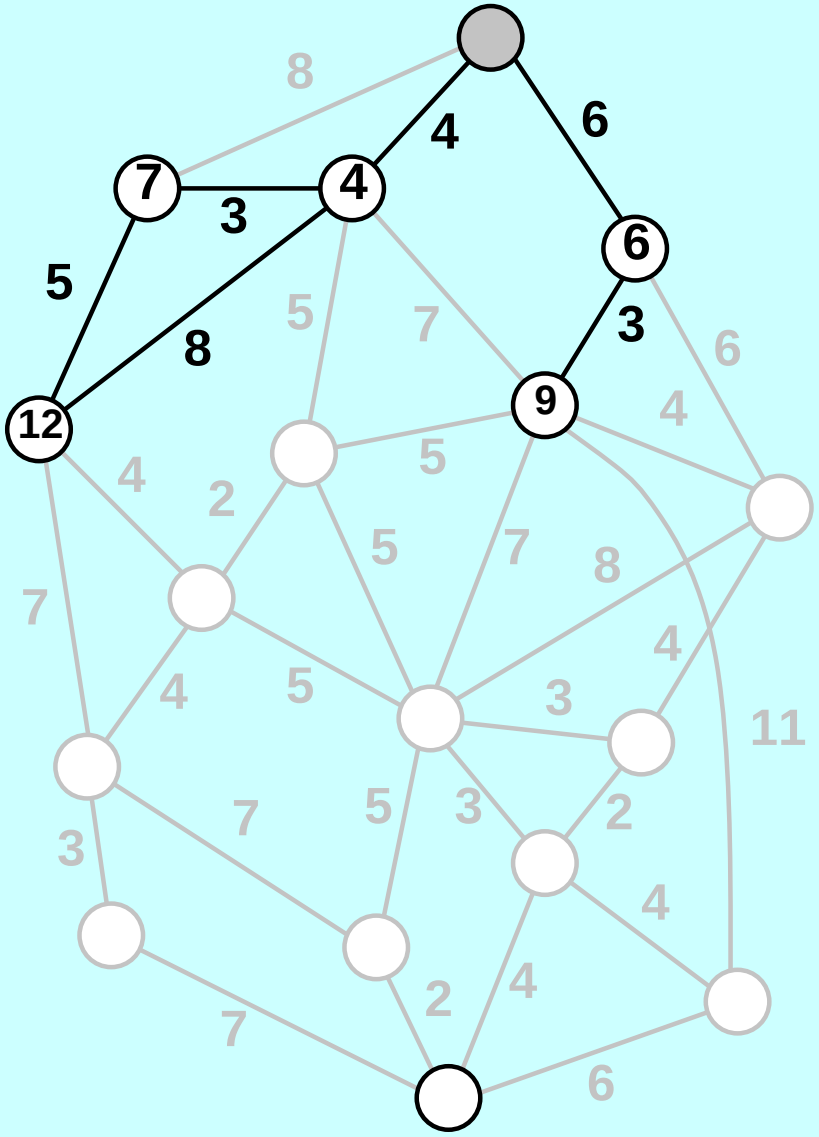
Shortest path through a graph



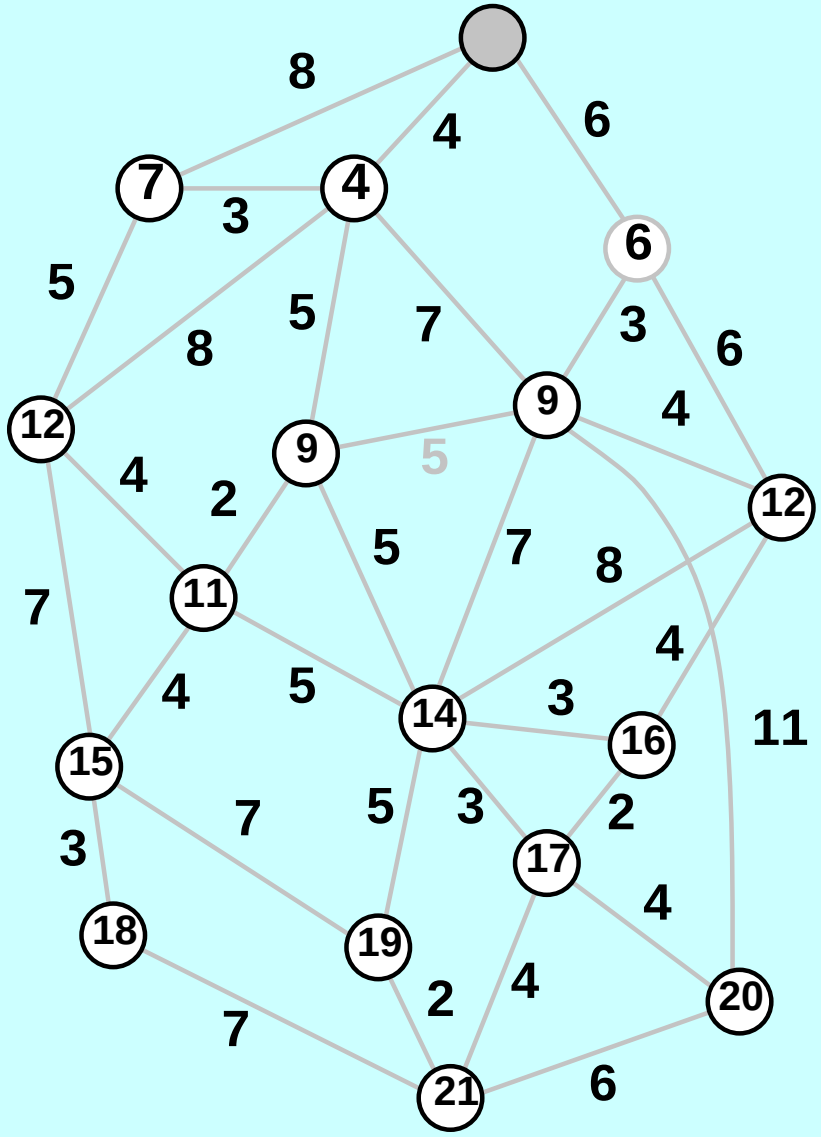
Shortest path through a graph



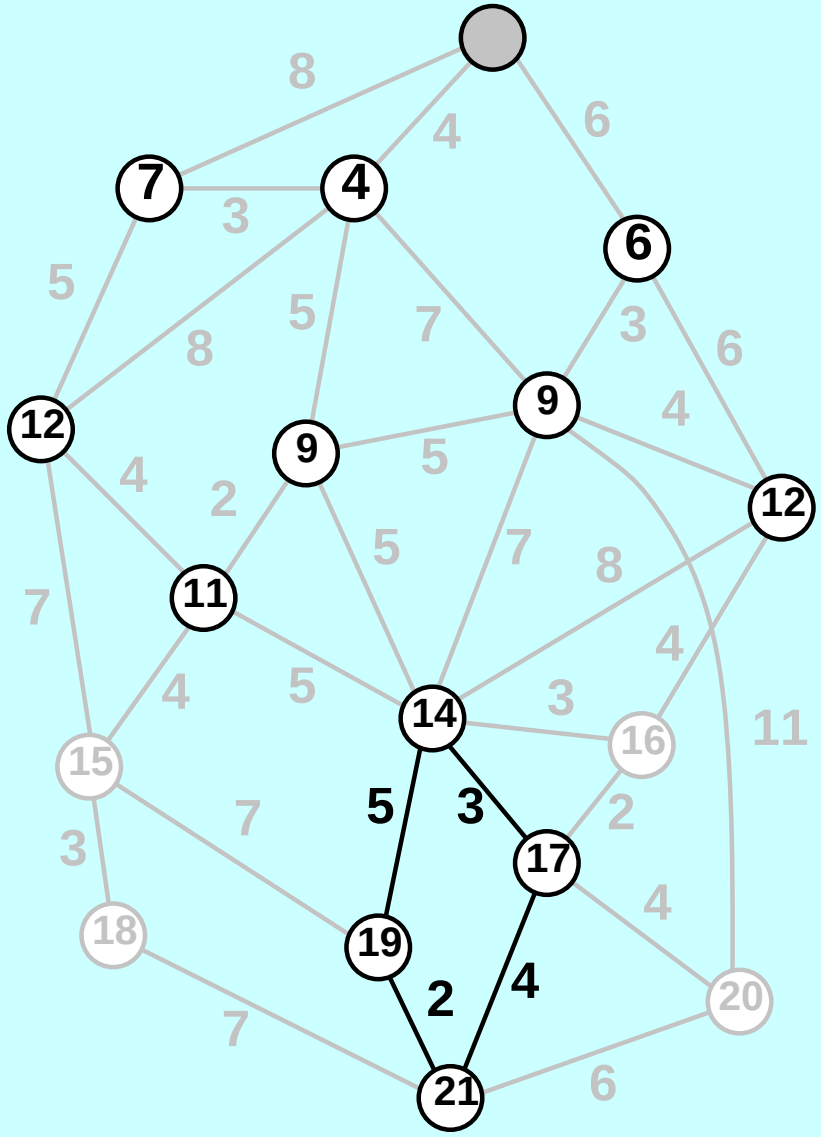
Shortest path through a graph



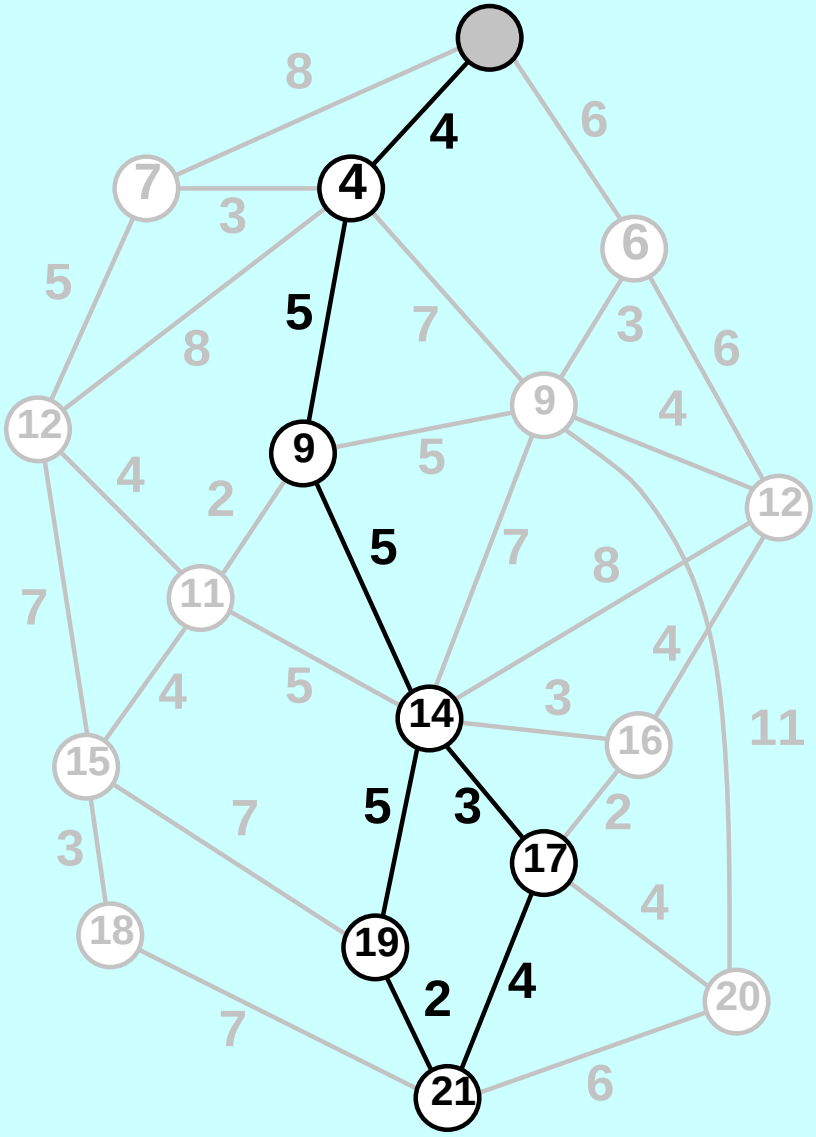
We finally get scores for all nodes



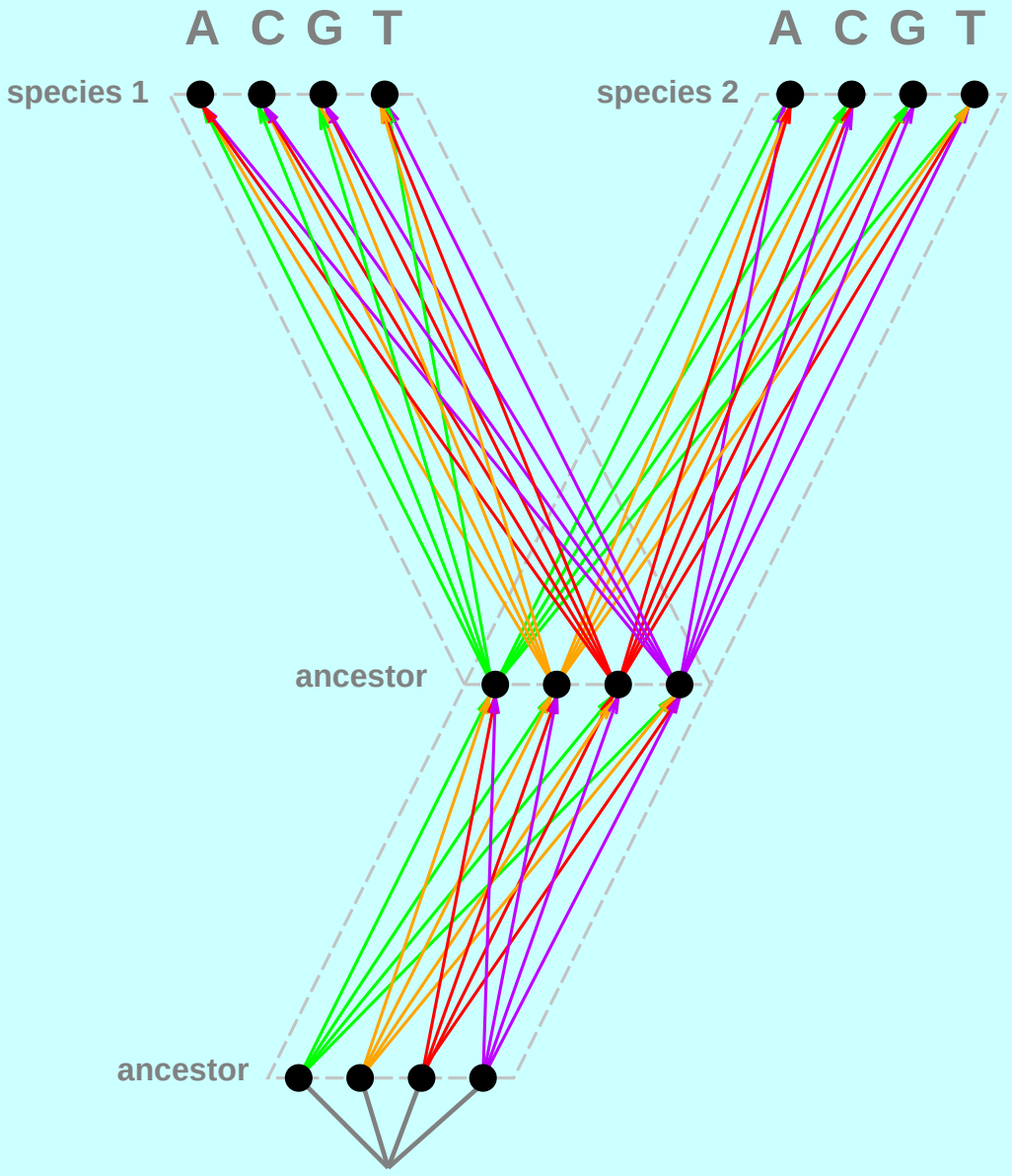
Then we go back from goal



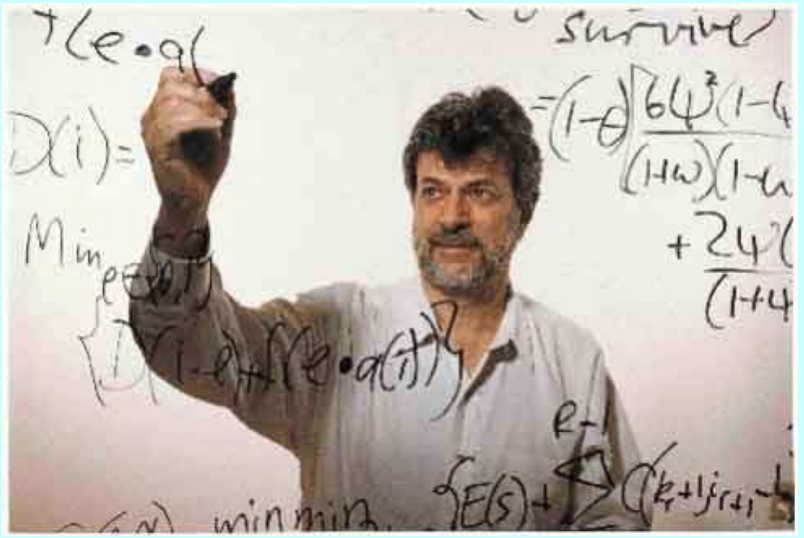
This is a ‘dynamic programming’ algorithm



Parsimony is a shortest-path problem

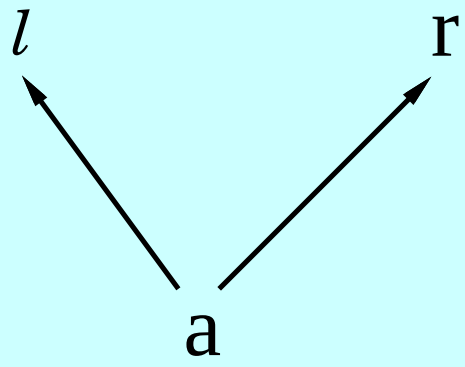


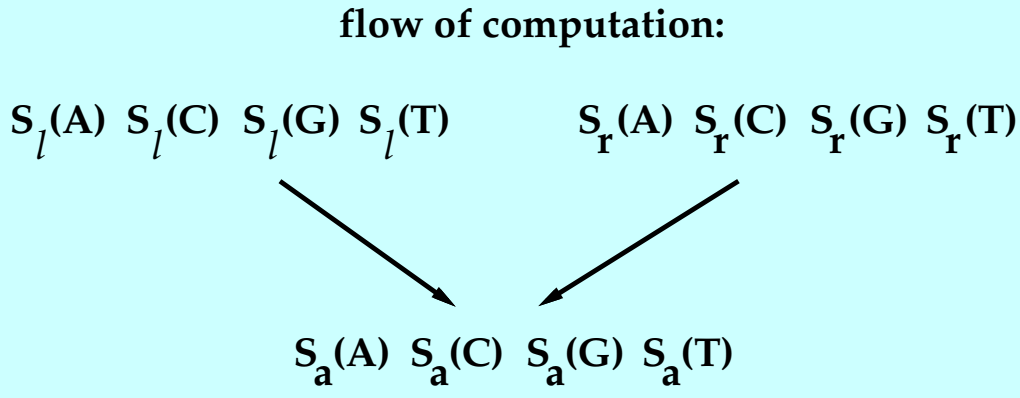
David Sankoff



David Sankoff, in the 1990s, writing on a glass panel
(forwards, then he went behind it)

The Sankoff algorithm





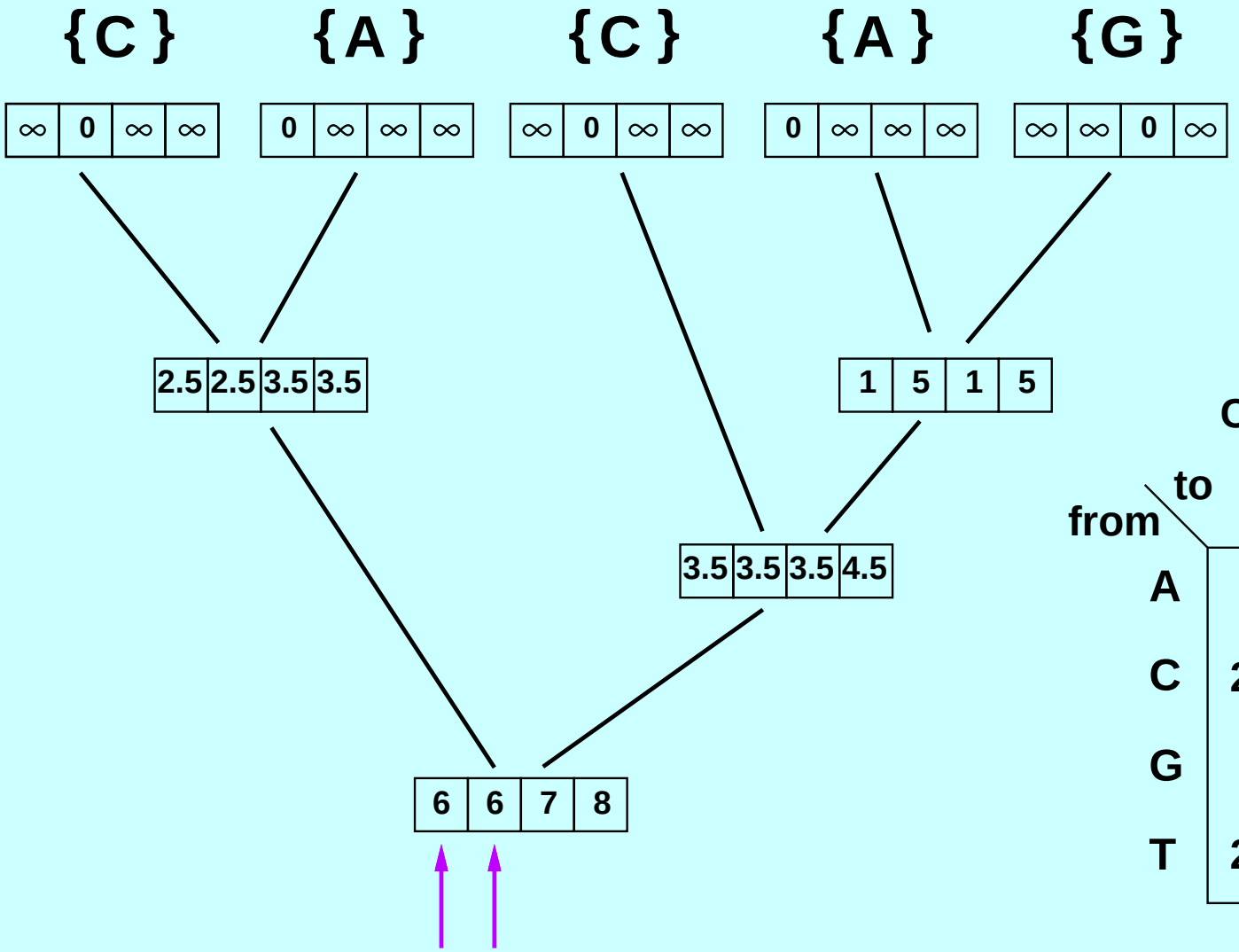
1. $S_k(i)$ is defined as the number of steps required at or above node k given that node k is in state i.
2. Set these quantities at the tips for the character (they are either 0 or ∞).
3. move down the tree doing this at each node:

$$S_a(i) = \min_j [c_{ij} + S_\ell(j)] + \min_k [c_{ik} + S_r(k)]$$

4. At the bottom node of the tree:

$$S = \min_i S_0(i)$$

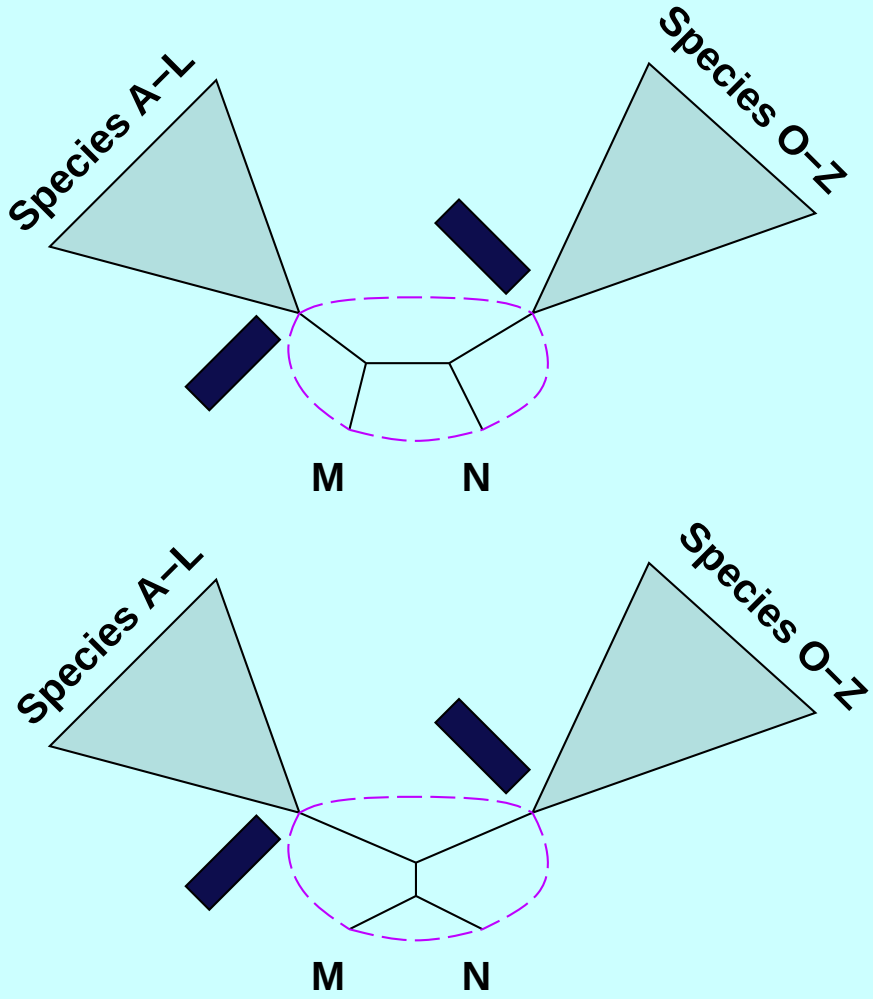
Example for the Sankoff algorithm



Cost matrix:

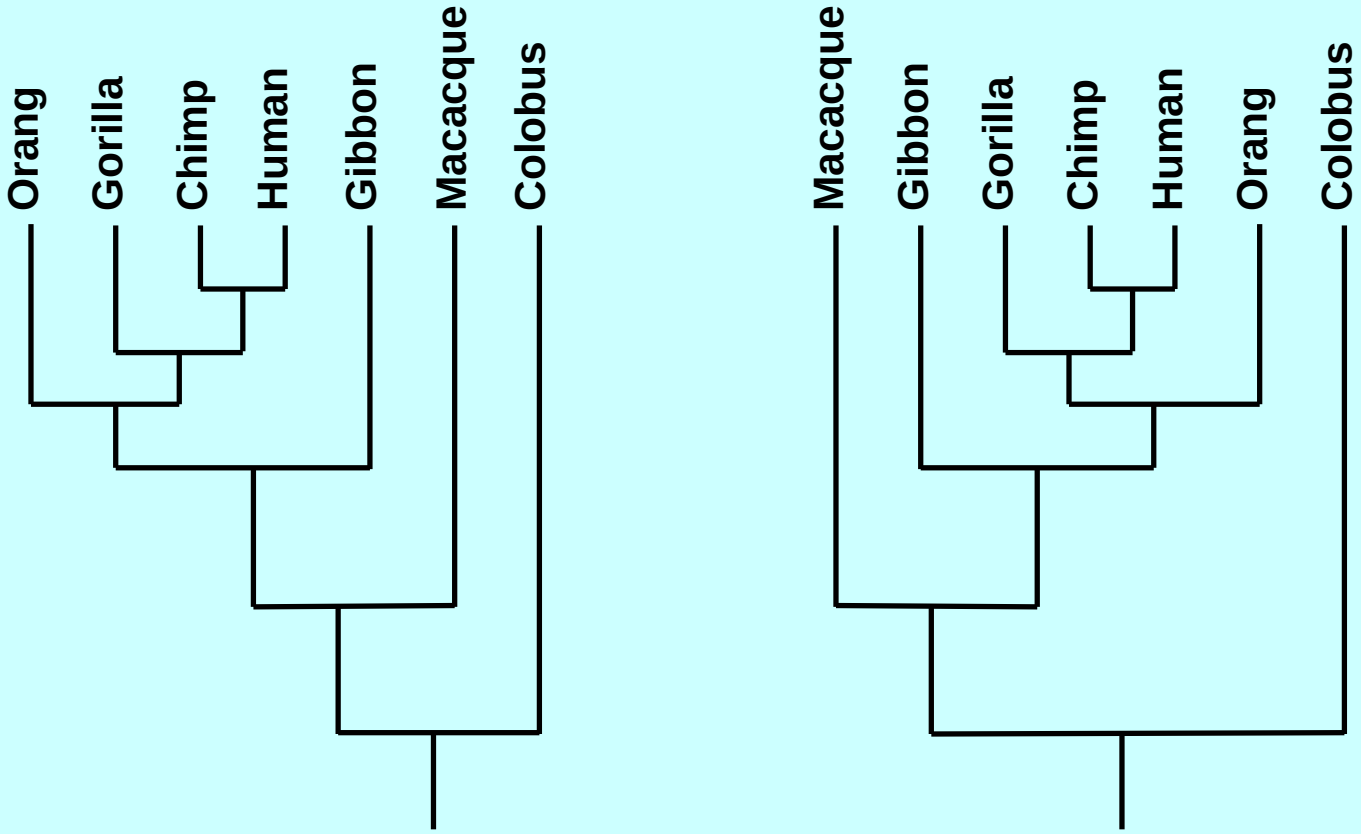
from \ to	A	C	G	T
A	0	2.5	1	2.5
C	2.5	0	2.5	1
G	1	2.5	0	2.5
T	2.5	1	2.5	0

Economizing on the score computation



By retaining the interior arrays of scores, we can avoid having to redo the parts of the tree that have not changed. The result is a great speedup if only a small part of the tree has changed.

The same tree?

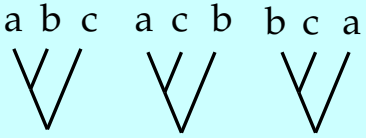


(It's important that you know how to answer this. Also that you be able to recognize whether two different rooted trees are the same unrooted tree).

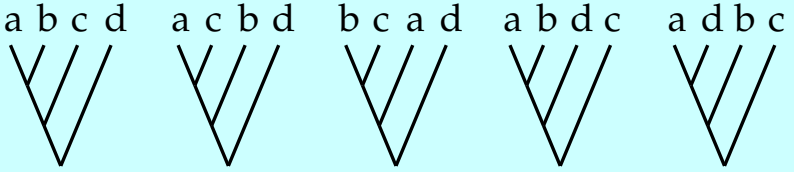
Rooted, labelled, bifurcating trees

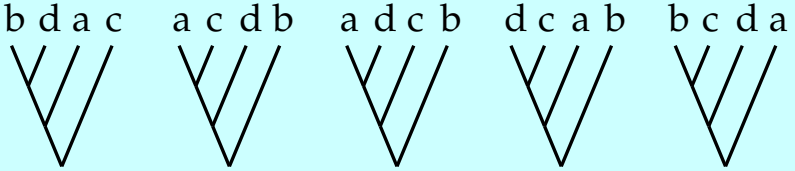
$n = 2$

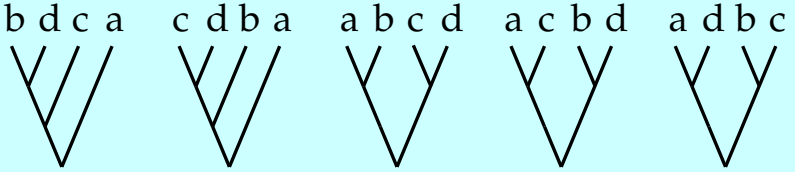
$n = 3$



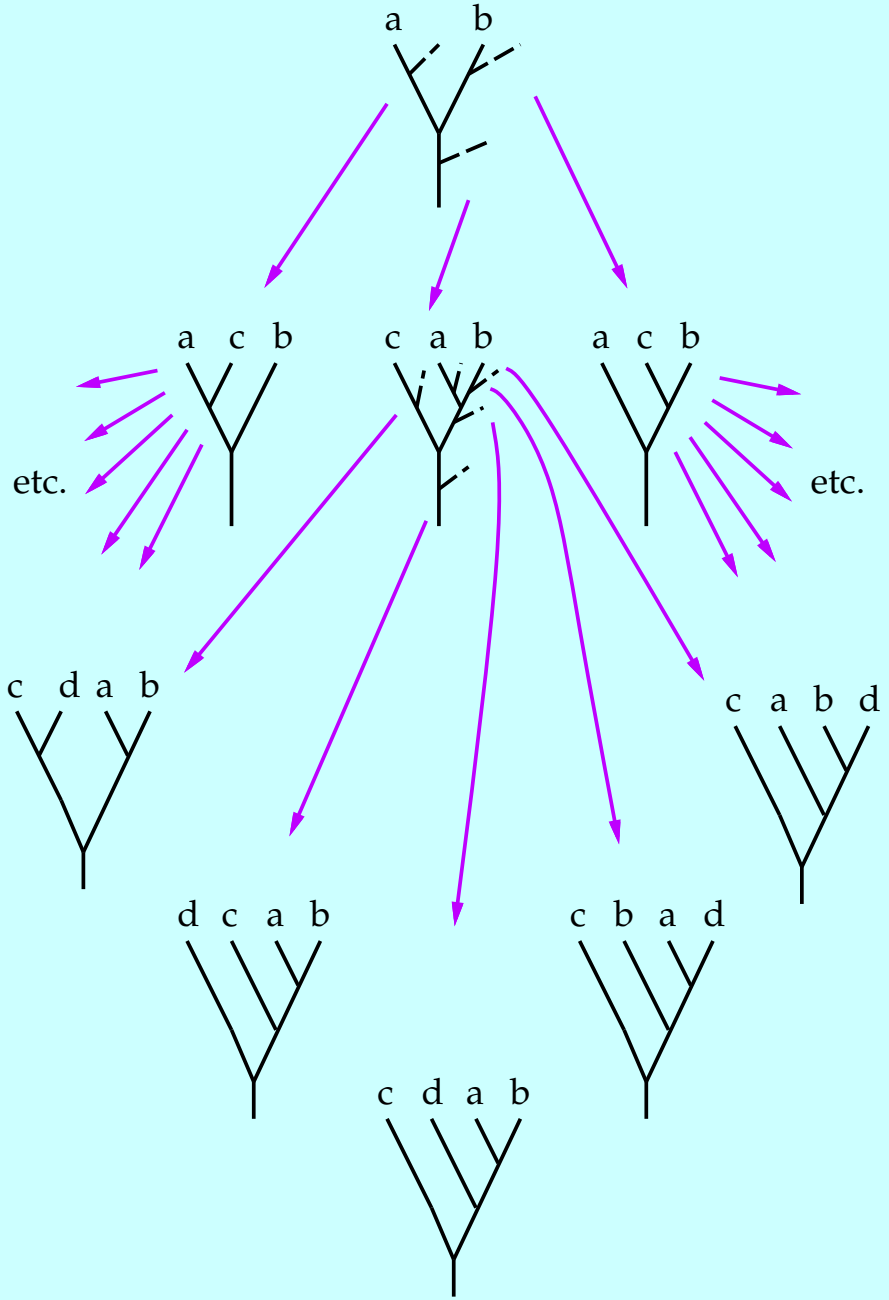
$n = 4$







Adding species in all possible places



The resulting count of the number of trees

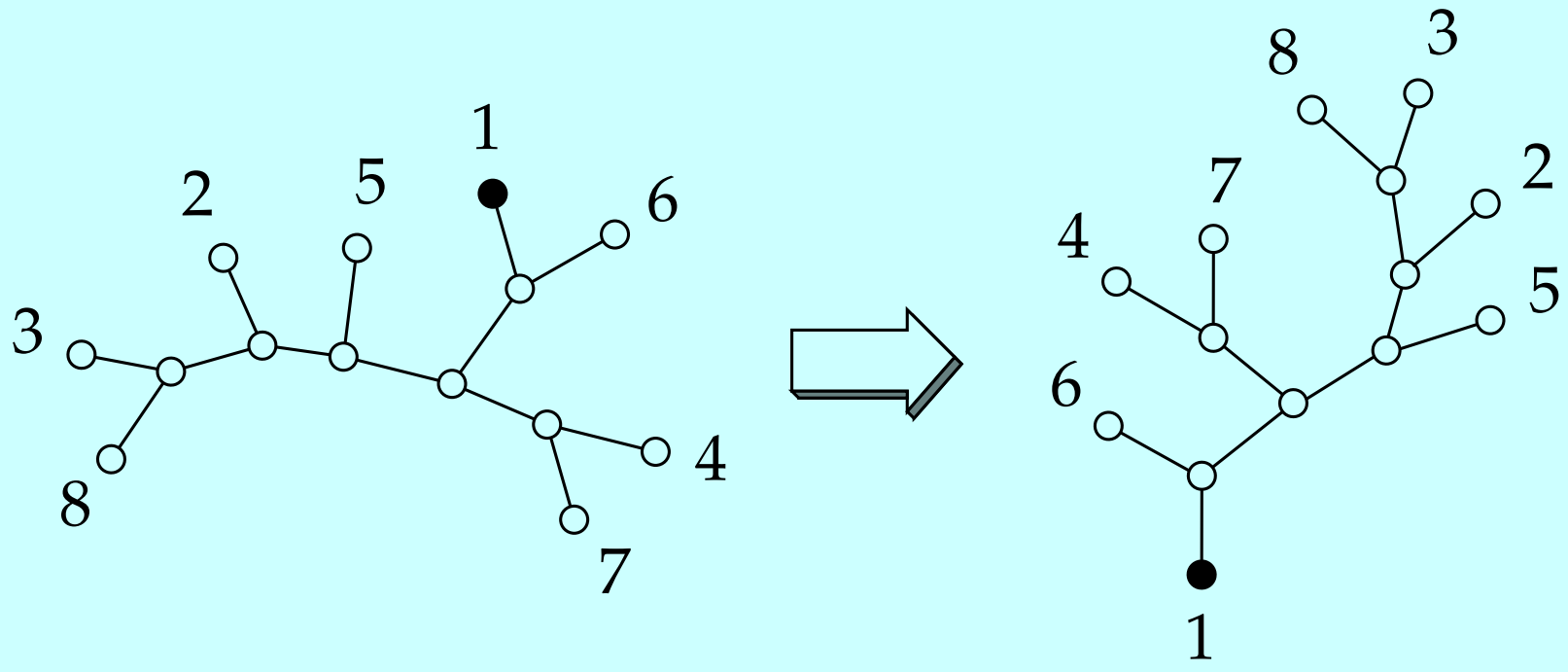
$$1 \times 3 \times 5 \times 7 \times 9 \times \ldots \times (2n - 3)$$

We can get this from the argument about adding species in a particular order (say, alphabetically) in all possible places on the tree. Once we realize that all rooted bifurcating trees with that set of species can be reached this way, and that each one can be reached in only one such way, it becomes obvious that the above product counts the number of possible trees.

Rooted, bifurcating, labelled trees

species	number of trees
1	1
2	1
3	3
4	15
5	105
6	945
7	10,395
8	135,135
9	2,027,025
10	34,459,425
11	654,729,075
12	13,749,310,575
13	316,234,143,225
14	7,905,853,580,625
15	213,458,046,676,875
16	6,190,283,353,629,375
17	191,898,783,962,510,625
18	6,332,659,870,762,850,625
19	221,643,095,476,699,771,875
20	8,200,794,532,637,891,559,375
30	4.9518×10^{38}
40	1.00985×10^{57}
50	2.75292×10^{76}

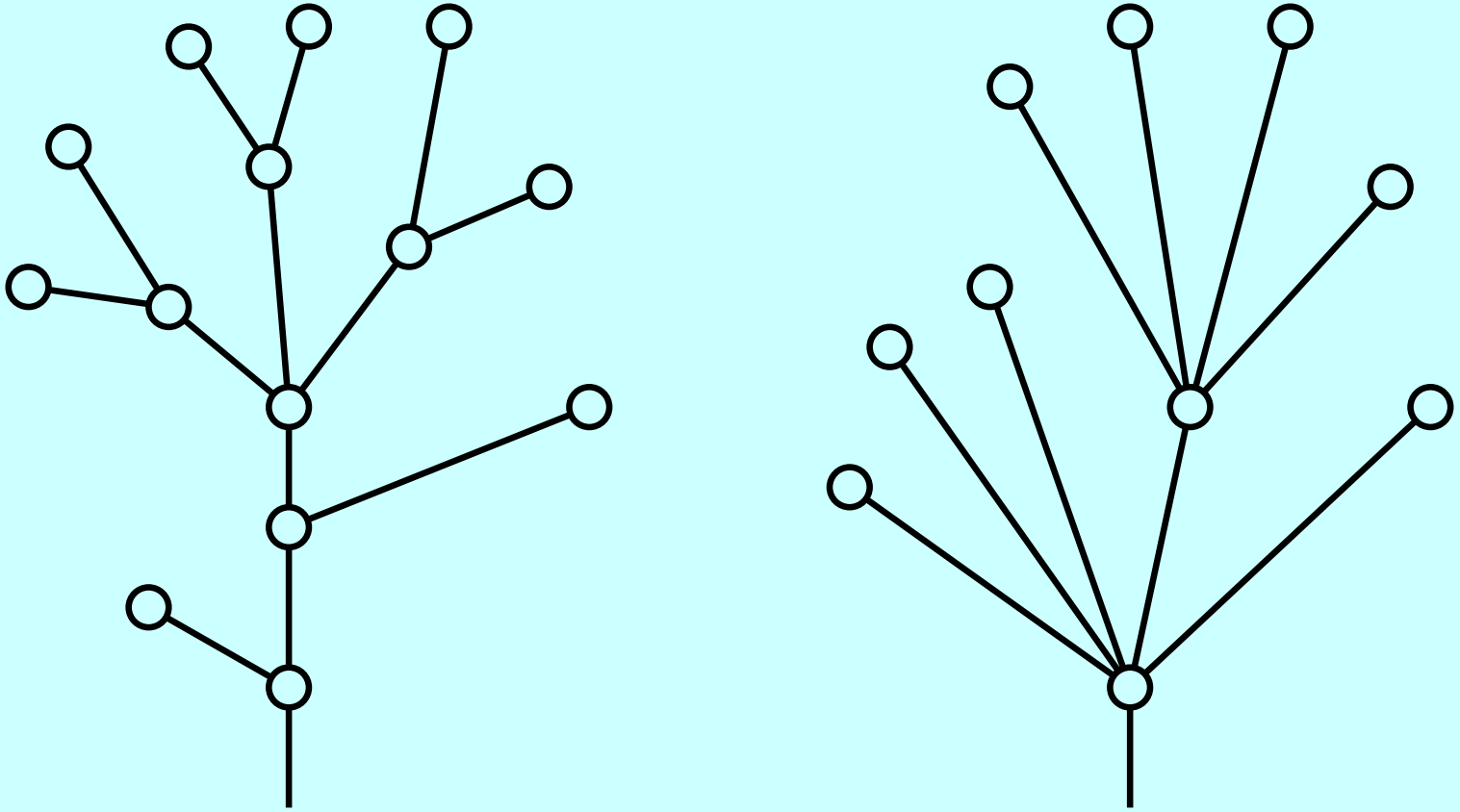
Rooting an unrooted tree at species 1



So an unrooted tree can be considered as being a rooted tree, provided one of the tip species becomes the “root” so that the rooted tree has one fewer species.

Thus the number of unrooted trees is the same as the number of rooted trees, ones that have one fewer species.

Multifurcating trees



Note that they have different numbers of branches as well. To count them we have to keep track of the numbers of trees with different numbers of interior nodes. Then it is fairly easy. I introduced this method in 1978; the number of rooted multifurcating trees was first counted 108 years earlier by Ernst Schröder in 1870 using generating function methods.

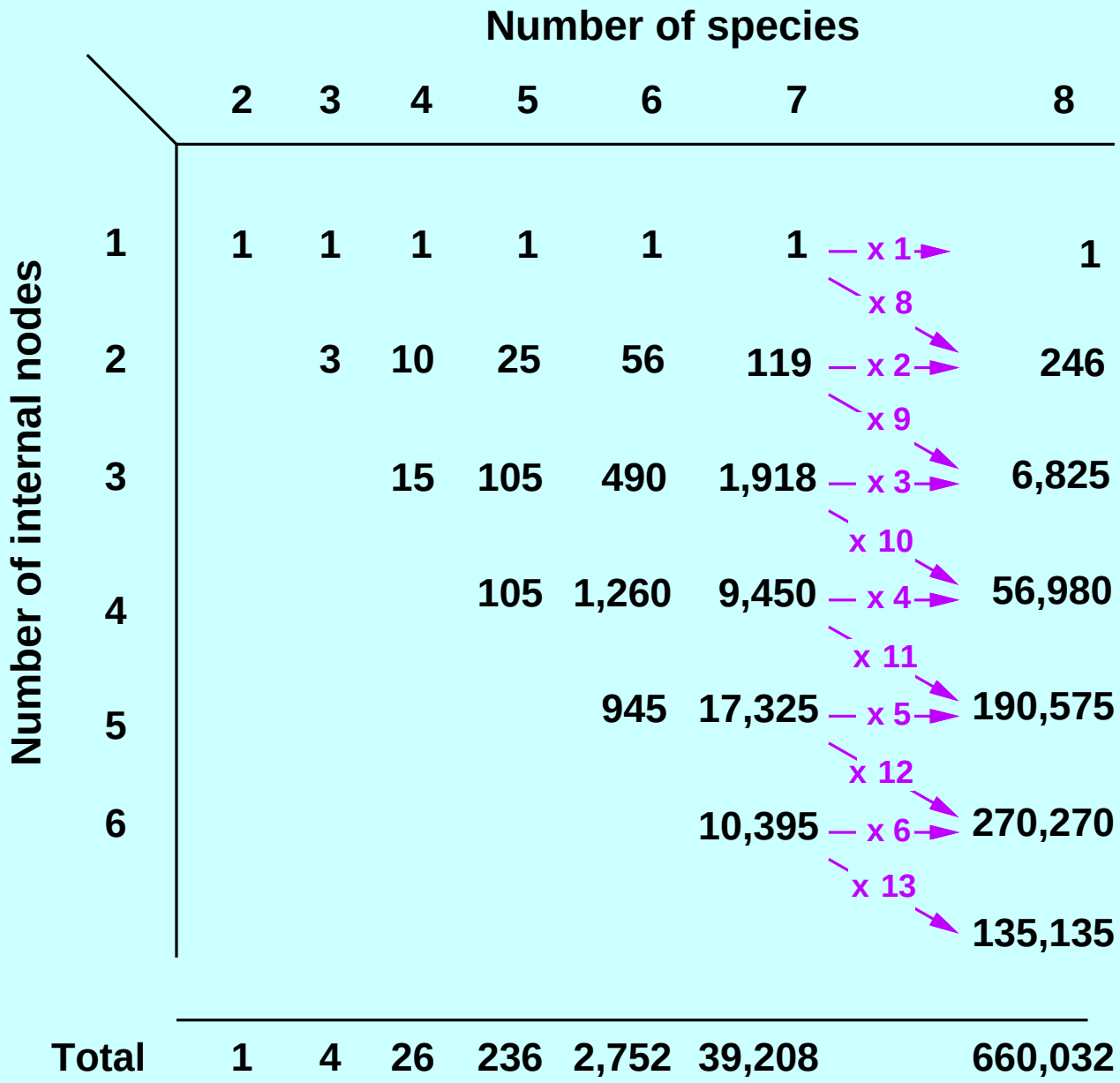
Counting multifurcating rooted trees

If we add a new species to a tree that has n tips and m interior nodes, there are two kinds of places we could add them:

1. Branching off of one of the $n + m$ branches.
2. Coming out of one of the m interior nodes.

Each of these generates a different new tree. Branching off of a branch creates a new interior node, coming out of an interior node does not create a new interior node.

The algorithm counting multifurcating rooted trees



Rooted trees allowing multifurcations

species	number of trees
2	1
3	4
4	26
5	236
6	2,752
7	39,208
8	660,032
9	12,818,912
10	282,137,824
11	6,939,897,856
12	188,666,182,784
13	5,617,349,020,544
14	181,790,703,209,728
15	6,353,726,042,486,272
16	238,513,970,965,257,728
17	9,571,020,586,419,012,608
18	408,837,905,660,444,010,496
19	18,522,305,410,364,986,906,624
20	887,094,711,304,119,347,388,416
30	7.0717×10^{41}
40	1.9037×10^{61}
50	6.85×10^{81}
100	3.3388×10^{195}

Wedderburn's algorithm for numbers of tree shapes

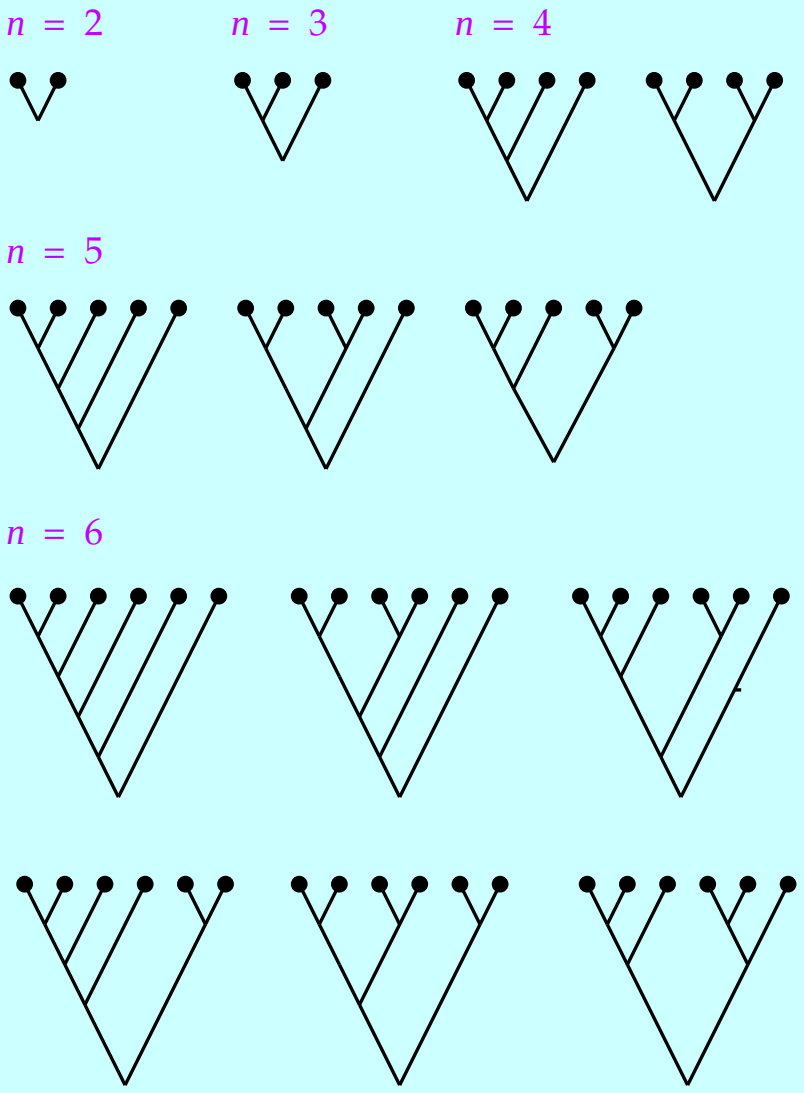
$$
\begin{aligned}
S_1 &= 1 \\
S_n &= S_1 S_{n-1} + S_2 S_{n-2} + \ldots + S_{(n-1)/2} S_{(n+1)/2} && \text{if } n > 1 \text{ and } n \text{ is odd} \\
S_n &= S_1 S_{n-1} + S_2 S_{n-2} + \ldots + S_{n/2}(S_{n/2} + 1)/2 && \text{if } n > 1 \text{ and } n \text{ is even}
\end{aligned}
$$

This algorithm is easily obtained by considering that a rooted bifurcating tree with n tips has a fork at its base with a tree of k tips on one side, and a tree of $n - k$ tips on the other, and that exchanging the order of these does not change the tree.

Shapes of rooted bifurcating trees

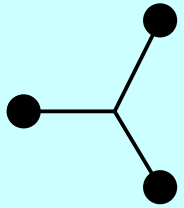
species	number of shapes
1	1
2	1
3	1
4	2
5	3
6	6
7	11
8	23
9	46
10	98
11	207
12	451
13	983
14	2,179
15	4,850
16	10,905
17	24,631
18	56,011
19	127,912
20	293,547
30	1.4068×10^{9}
40	8.0997×10^{12}
50	5.1501×10^{16}
100	1.0196×10^{36}

Rooted, bifurcating tree shapes

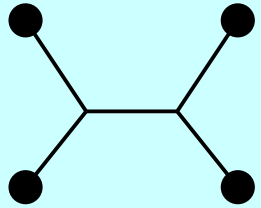


Unrooted bifurcating tree shapes

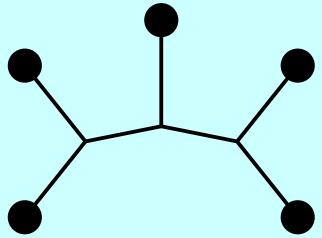
$n = 3$



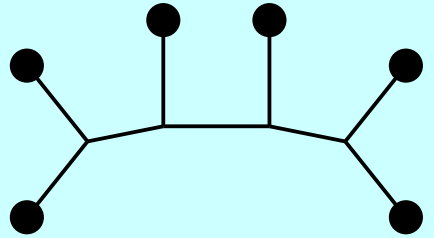
$n = 4$

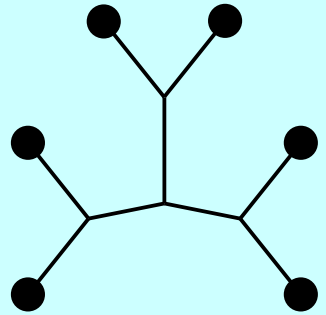


$n = 5$



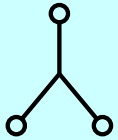
$n = 6$



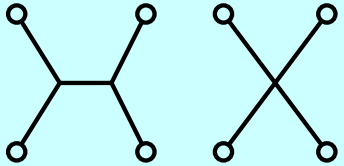


Unrooted multifurcating tree shapes

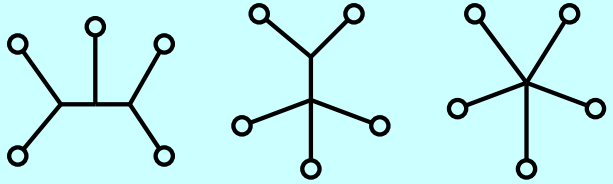
$n = 3$



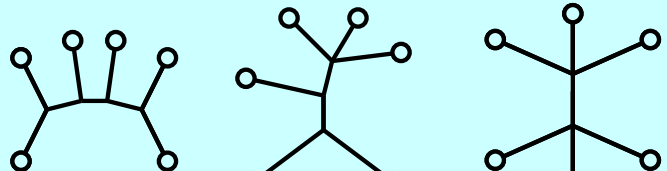
$n = 4$

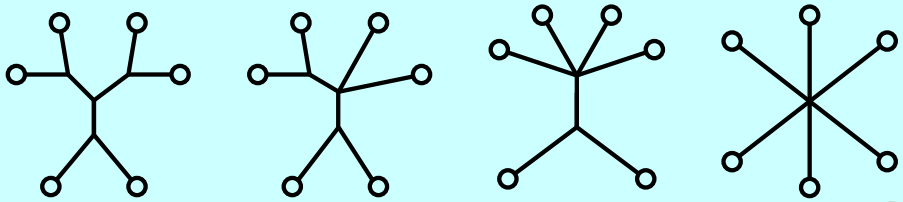


$n = 5$

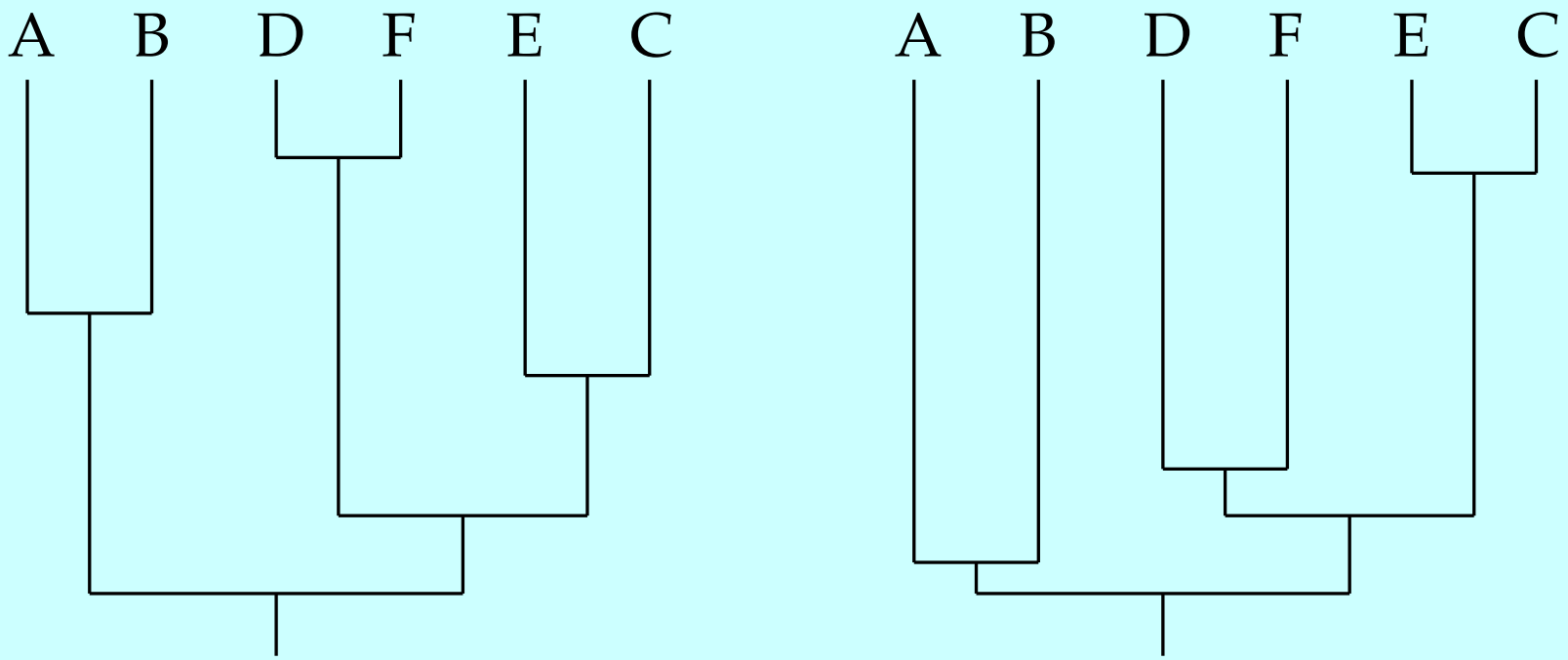


$n = 6$





Same topology but different labelled histories



Counting labelled histories

You can count labelled histories by going back in time, noting that there are $\binom{n}{2} = n(n-1)/2$ different pairs that can join most recently. Before that there are $n-1$ lineages so there are $(n-1)(n-2)/2$ pairs that can join. Going all the way back to the root, just before it $2 \times 1/2$ pairs can join.

Each combination of choices leads to a different labelled history, so we take the product of these numbers.

This leaves us with

$$\frac{n!(n-1)!}{2^{n-1}}$$

labelled histories.

The number of labelled histories

n	Number
2	1
3	3
4	18
5	180
6	2700
7	56700
8	1587600
9	57153600
10	2571912000
11	141455160000
12	9.336041×10^{12}
13	7.282112×10^{14}
14	6.626722×10^{16}
15	6.958058×10^{18}
16	8.349669×10^{20}
17	1.135555×10^{23}
18	1.737399×10^{25}
19	2.970953×10^{27}
20	5.644810×10^{29}