1. The dissimilarity map δ for A, B, C, and D is shown below. Find an example to show that δ is *not* an ultrametic. Then find the "closest" ultrametic to δ.

δ	A	B	C	D
A	0	6	4	2
B	6	0	1	3
C	4	1	0	5
D	2	3	5	0

Solution: (sketch) We can select labels A,B, and C to show that δ is not an ultrametric:

$$\delta(A, B) = 6 \not\leq \max\{\delta(A, C), \delta(C, B)\} = \max\{4, 1\}.$$

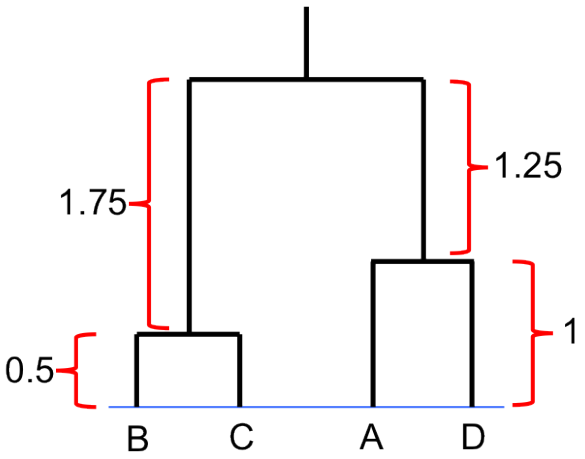
Now we can use UPGMA to find the "closest" ultrametric to δ, since the induced dissimilarity map from the UPGMA tree will be an ultrametric. The first step of UPGMA is to find the minimum distance between two leaf labels, which is $\delta(B, C) = 1$ in this case. We create a node whose children are labeled B and C, and define the height of this node as $1/2$. Now we need to create a new dissimilarity map Δ on the cluster $\{B, C\}$ and the labels A and D:

Δ	A	$\{B, C\}$	D
A	0	5	2
$\{B, C\}$	5	0	4
D	2	4	0

The next minimum distance is $\delta(A, D) = 2$, so we again merge these two labels into a cluster and give the resulting node a height of 1. The updated distance metric is now:

Δ	$\{B, C\}$	$\{A, D\}$
$\{B, C\}$	0	4.5
$\{A, D\}$	4.5	0

Note that although for this example we were averaging distances between clusters of the same size, in general this will not be the case and we'll need to weight by the number of leaves in each cluster. Finally we can merge the clusters $\{B, C\}$ and $\{A, D\}$ to arrive at our final tree:



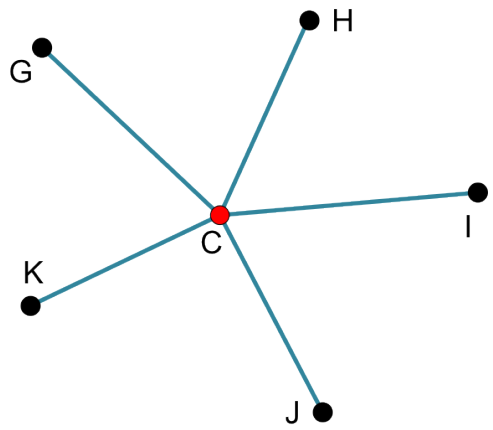
Finally, we can read off the induced ultrametric δ' on A, B, C, and D:

δ'	A	B	C	D
A	0	4.5	4.5	2
B	4.5	0	1	4.5
C	4.5	1	0	4.5
D	2	4.5	4.5	0

2. The dissimilarity map d below for G, H, I, J, and K is a *tree metric*. Find a tree $\mathcal{T}$ and edge weights w associated with d.

d	G	H	I	J	K
G	0	5	6	8	7
H	5	0	5	5	6
I	6	5	0	8	3
J	8	5	8	0	9
K	7	6	3	9	0

Solution: (sketch) Because neighbor joining is consistent (i.e. if we input a tree metric, we'll get back a tree with an induced metric that is the same as our input), we can use this algorithm to create an edge-weighted tree for this set of labels. First we begin with a star-shaped tree with a center node c:



Since c has degree 5, we need to perform 2 steps of neighbor joining to reduce its degree to 3. These steps are shown below.

Step 1: (a) First we compute the Q function for all $n = 5$ nodes attached to c (we call this set N_c). This requires computing S_i, which is the sum of the distances from i to all the other nodes attached to c, so $S_i = \sum_{k \in N_c} d(i,k)$.

i	G	H	I	J	K
S_i	26	21	22	30	25

Then we can compute $Q(i,j)$ for distinct $i, j \in N_c$:

$$Q(i,j) = (n-2)d(i,j) - S_i - S_j.$$

Q	G	H	I	J	K
G	0	-32	-30	-32	-30
H	-32	0	-28	-36	-28
I	-30	-28	0	-28	$\boxed{-38}$
J	-32	-36	-28	0	-28
K	-30	-28	$\boxed{-38}$	-28	0

The minimum value is $Q(I, K)$, so we will join these two nodes to make a cherry.

(b) Let u be the new parent node of I and K. The next step is to find $d(u, I)$ and $d(u, K)$:

$$d(u, I) = \frac{1}{2}d(I, K) + \frac{1}{2(n-2)}[S_I - S_K] = \frac{3}{2} - \frac{1}{2} = 1$$

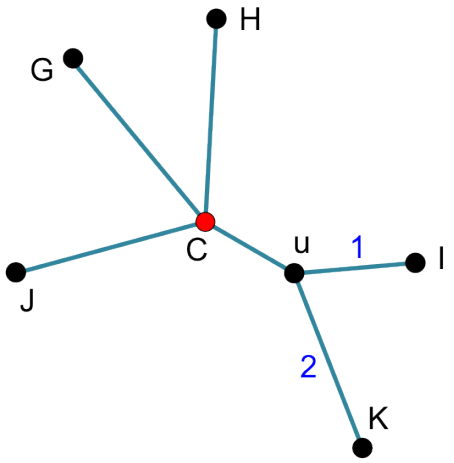
$$d(u, K) = \frac{1}{2}d(I, K) + \frac{1}{2(n-2)}[S_K - S_I] = \frac{3}{2} + \frac{1}{2} = 2$$

(c) The final step of an iteration of neighbor joining is to compute the distance from u to each of the other remaining nodes in N_c to create a new 4×4 distance matrix. For example:

$$d(u, G) = \frac{1}{2}[d(G, I) - d(I, u)] + \frac{1}{2}[d(G, K) - d(K, j)] = 5$$

d	u	G	H	J
u	0	5	4	7
G	5	0	5	8
H	4	5	0	5
J	7	8	5	0

So our resulting distance matrix from this iteration is above, and the tree is below.



Step 2: (a) We perform the same three sub-steps for the second iteration, with $n = 4$ nodes in N_c, first computing S and the Q function:

i	u	G	H	J
S_i	16	18	14	20

Q	u	G	H	J
u	0	-24	-22	-22
G	-24	0	-22	-22
H	-22	-22	0	$\boxed{-24}$
J	-22	-22	$\boxed{-24}$	0

Here we have two pairs with the minimum Q value, so we will arbitrarily choose to make a cherry out of H and J (it will not end up mattering for the tree topology).

(b) Let v be the new parent node of H and J. The next step is to find $d(v, H)$ and $d(v, J)$:

$$d(v,H) = \frac{1}{2}d(H,J) + \frac{1}{2(n-2)}[S_H - S_J] = \frac{5}{2} - \frac{3}{2} = 1$$

$$d(v,J) = \frac{1}{2}d(H,J) + \frac{1}{2(n-2)}[S_J - S_H] = \frac{5}{2} + \frac{3}{2} = 4$$

(c) Finally we can find the distances from v to the remaining nodes in N_c:

d	u	v	G
u	0	3	5
v	3	0	4
G	5	4	0

After this process we have constructed the correct topology (now c has degree 3, which is what we want for a binary tree). But there are still three edge weights we don't know, corresponding to the three edges attached to c. But we also have three pieces of information from the final distance matrix above. We can use these to solve for the remaining edge weights to obtain $d(c, u) = 2$, $d(c, v) = 1$, and $d(c, G) = 3$. Thus our final tree is: