**Week 1**

1. True or False: according to its definition, a distance matrix must satisfy the triangle inequality.

**True**

False

1. True or False: according to its definition, a distance matrix must have all positive values.

True

**False**

1. True or False: most distance matrices constructed from biological data are non-additive.

**True**

False

1. How many edges must a tree with 2349 nodes have?

**2348** [edges = nodes - 1]

1. How many nodes must a tree with 1167 edges have?

**1168** [nodes = edges + 1]

1. Which of the following matrices is fit by the tree shown below?

A diagram of a molecule

AI-generated content may be incorrect.

|  |  |  |  |
| --- | --- | --- | --- |
| i j k l  i 0 6 10 13  j 6 0 12 15  k 10 12 0 7  l 13 15 7 0 | i j k l  i 0 6 9 14  j 6 0 11 16  k 9 11 0 7  l 14 16 7 0 | i j k l  i 0 6 10 14  j 6 0 12 16  k 10 12 0 6  l 14 16 6 0 | **i j k l**  **i 0 6 9 13**  **j 6 0 11 15**  **k 9 11 0 6**  **l 13 15 6 0** |

1. Compute *LimbLength*(*i*) for the additive distance matrix shown below.

i j k l

i 0 13 16 10

j 13 0 21 15

k 16 21 0 18

l 10 15 18 0

**4**

1. Compute *LimbLength*(k) for the additive distance matrix shown below.

i j k l

i 0 14 17 17

j 14 0 7 13

k 17 7 0 16

l 17 13 16 0

**5**

**Week 2**

1. True or False: UPGMA does not necessarily identify neighboring leaves in the tree fitting an additive distance matrix D.

**True**

False

1. Compute the sum of squared errors *Discrepancy*(*T*, *D*) for the tree *T* and distance matrix *D* given below.

A diagram of a network

AI-generated content may be incorrect.i j k l

i 0 20 9 11

j 20 0 17 11

k 9 17 0 8

l 11 11 8 0

**8**

|  |  |
| --- | --- |
| D = [[0, 20, 9, 11],  [20, 0, 17, 11],  [9, 17, 0, 8],  [11, 11, 8, 0]] | T = [[0, 20, 8, 12],  [20, 0, 18, 12],  [8, 18, 0, 10],  [12, 12, 10, 0]] |

3. Below is a distance matrix *D*. If *C*1 is the cluster containing *i* and *j*, and *C*2 is the cluster containing *k* and *l*, compute *D*(*C*1, *C*2).

i j k l

i 0 20 9 11

j 20 0 17 11

k 9 17 0 8

l 11 11 8 0

**12**

4. Below is a distance matrix *D*. Compute *D\*k, l* where *D\** is the neighbor-joining matrix of *D*.

i j k l

i 0 13 16 10

j 13 0 21 15

k 16 21 0 18

l 10 15 18 0

**-62**

5. Below is a distance matrix *D*. After the neighbor-joining algorithm decides that *j* and *l* are neighbors, compute *LimbLength*(*l*).

i j k l

i 0 20 9 11

j 20 0 17 11

k 9 17 0 8

l 11 11 8 0

**1**

Week 3