Qn – 1:

For each pair of 3D tree images i and j the scientists look at them side by side and decide whether they are the "same" species or "different". They also have the option of not giving an opinion and just leave the pair without a decision. So now the scientists have the collection of n trees, collection of m decisions (either "same" or "different") for the pairs for which some decision was made.

Give an algorithm with running time O(m + n) that determines whether the m decisions are consistent. Don't forget to prove its correctness and termination. Note that the input consists of the number of trees, n, and a list of m decisions for some pairs of trees

Algorithm:

Step 1: Create a graph G where trees are vertices and the decisions are edges (irrespective whether the decision was same or different)

So, the Graph will have n vertices and m edges.

Step 2: Now, perform a breadth first search on the tree starting from any arbitrary node and label the trees A or B based on the decision between the edges

Let C be list of all explored nodes, initially C = {s} where s is the arbitrarily chosen node

L[0] = {vertices with direct edge to s}

Let "A" and "B" be two possible labels

Label s as "A"

i = 0

While L[i] is not empty:

For each edge {u, v} where u belongs to C and v not in C:

If Decision b/w {u, v} was "same" then label v with same label as u

If Decision b/w {u, v} was "different" label v with opposite label of u

Add v to C

L[i + 1] = {vertices with edges from L[i] and not in C}

i = i + 1

If there is still any vertex which is unexplored. Assign the vertex as the arbitrary start node and continue and run the algorithm. Repeat this till all vertices are explored and have a label

Step 3: Check if the Labelling done using BFS is consistent or not.

For each pair of decisions {u, v},

if Decision {u, v} == same and u, v have different label then

set Decisions as inconsistent

Break

Else if Decision {u, v} == different and u, v have same label.;

set Decisions as inconsistent

Break

Else

set Decision as consistent

Continue loop

Proof of Correctness:

Scenario – 1: Case when BFS produces inconsistent labelling when there are no inconsistencies in the decisions

1. The labelling is done using BFS from looking at the decision between two adjacent nodes in the graph and the label is assigned either as same to its neighbor or different to its neighbor based on the decision between the pairs.
2. Hence if there is some inconsistency with the labelling from BFS then we can be sure that the complete set of m decisions cannot consistent with on another
3. **Hence this scenario cannot occur**

Scenario – 2: Case when BFS produces consistent labelling when there are inconsistencies in the decisions

1. Since we label a node only once based on the first time it is traversed and its relationship with its neighbor, If there exists another decision which contradicts with a previous decision involving the node – the node is not relabeled
2. Hence when the labeling is checked post BFS with each decision, the inconsistency will be identified when this decision is visited.
3. **Hence this scenario also cannot occur**

Proof of Termination:

Since the nodes are visited only once and no previously explored node is visited again it will terminate at most (n + m) iterations

Complexity:

Creating the graph – O(n + m)

Traversing the graph (BFS) – O(n + m)

Checking if the labels from BFS are consistent with the decisions – O(n)

Total time complexity is O(n + m)

Extra Credit:

This algorithm will not work for cases where there are more than 2 species as it involves having only 2 labels and giving the neighboring node either the same label or opposite label based on decision.

But if more than 2 labels are present, for instance 3 labels then if decision between two trees is different – there are two viable options to name it

Consider following decisions, {1,2} is different, {2,3} is different, {1,3} is different, {2,4} is different.

If number of species = 2: Say we choose 1 as starting node – 1 is labelled as "A"

2 – "B", 3 – "B", 4 – "A"

But we know {2,3} cannot have same label and hence we can call these decisions inconsistent

If number of species = 3: According this algorithm, say we choose 1 as starting node

1 – "A", 2 – "B" or "C", 3 – "B" or "C" and there is no way to label 4 as it can be "A" or "B" or "C"