SCIENCE 2 - ASSIGNMENT 5
(MSA & Phylogeny)

ROMICA RAISINGHANI
2021101053

Ans1) Multiple sequence alignment is a fundamental bioinformatics technique used to compare and analyze multiple sequences simultaneously. The technique involves aligning three or more sequences of nucleotides or amino acids to identify conserved regions and variations among the sequences. The resulting alignment can then be used to infer ~~reveal~~ evolutionary relationships, identify conserved motifs, and predict functional sites.

Some applications of multiple sequence alignment are:

1) Phylogenetic analysis: Multiple sequence alignment is commonly used in evolutionary biology to infer phylogenetic relationships between species. By comparing the differences and similarities in multiple sequences, researchers can construct phylogenetic trees to visualize the evolutionary relationships among species.

2) Structure Prediction: Multiple sequence alignment is also used to predict the three-dimensional structure of proteins. By identifying conserved regions among protein sequences, researchers can infer the structural and functional properties of unknown proteins.

3) Functional annotation: Multiple sequence alignment can also be used in to identify conserved motifs and functional domains among sequences. These conserved regions can provide insights into the biological functions of proteins.

4) SNP analysis: Multiple sequence alignment is often used to analyze single nucleotide polymorphisms (SNPs) among multiple sequences. By identifying SNPs and their locations within a gene, researchers can infer the evolutionary history of the gene and its potential role in disease.

Now, regarding whether a multiple alignment carries more information than a pairwise alignment, the answer is generally yes. Multiple alignments can capture more information about the evolutionary history of the sequences and their relationships than pairwise alignment. This is because multiple alignments can reveal the patterns of conservations and variation among multiple sequences, while pairwise alignments only compare two sequences at a time. Multiple alignments can also help identify regions of functional importance that might be missed in pairwise alignment.

Ans2) The sum-of-pairs (SP) score is a commonly used measure of the quality of a multiple sequence alignment. The SP score is calculated by summing up the scores of all possible pairwise alignments between the sequences in the multiple alignment. Each pairwise alignment is assigned a score based on a scoring matrix that reflects the similarity between the aligned residues.

The SP score has several drawbacks that should be considered:

1) It is not additive: The SP score is not additive, which means that the score of a multiple alignment

cannot be computed by adding up the scores of its constituent pairwise alignments. This can make it difficult to compare the quality of different multiple alignments.

(2) It ignores gaps: The SP score does not take into account the presence of gaps in the multiple alignment. This can be problematic because gaps can have a significant impact on the functional properties of a protein.

(3) It is sensitive to sequence length: The SP score is sensitive to the length of the sequences being aligned. This means that the score of a multiple alignment may be biased towards longer or shorter sequences.

To address these issues, alternative scoring systems have been developed. One such alternative is the total-column score (TC score), which sums up the scores of all columns in the multiple alignment. The TC score is additive and takes into account gaps, making it less sensitive to sequence length than the SP score. Another alternative is the maximum expected accuracy (MEA) score, which uses a probabilistic model to calculate the expected accuracy of the alignment. The MEA score takes into account the probability of alignment errors and can be more accurate than the SP score.

Ans3)- The progressive alignment approach is a commonly used method for constructing multiple sequence alignments.
The basic steps involved in this approach are as follows:

(1) Pairwise alignment: The first step is to perform a pairwise alignment of the sequences using a scoring matrix. The alignment is then represented as a guide tree, which reflects the evolutionary relationships between the sequences.

(2) Tree-building: The guide tree is then used to guide the progressive alignment of the sequences. The tree is built by clustering the sequences based on their similarity scores.

(3) Multiple sequence alignment: The sequences are then progressively aligned from the leaves to the root of the tree. The alignment starts with the two sequences that are closest to each other in the tree, and then the remaining sequences are added one by one. At each step, the sequences are aligned to the existing alignment by using dynamic programming algorithms.

(4) Refinement: Finally, the alignment is refined by using an iterative process. This involves realigning regions of the alignment that are poorly aligned and adjusting the gaps in the alignment.

The progressive alignment approach has several drawbacks that should be considered:

(1) Sensitivity to initial pairwise alignment: The accuracy of the

final alignment can depend on the quality of the initial pairwise alignment. If the initial alignment is poor, then the final alignment may also be poor.

② Lack of global optimization: The progressive alignment approach uses a local optimization strategy, which means that it can get stuck in local optima and may not produce the globally optimal alignment.

③ Inability to handle structural information: The progressive alignment approach does not take into account structural information, which can be important for proteins that have conserved secondary and tertiary structures.

To overcome these shortcomings, several modifications have been made to the progressive alignment approach.
For example, iterative refinement ~~approach~~ methods can be used to improve the accuracy of the alignment. Alternatively, structural information can be incorporated into the alignment by using structural alignment algorithms or by using homology modeling techniques to predict the structure of the aligned sequences.

Ans 4) -

Given: Sequences are L = 50 residues long.

Pairwise alignment of two sequences takes $(2L)^{N-2} = 10^{2N-4} = 10^4$ seconds.

Now, if we had unlimited memory and were willing to wait for the answer until just before the sun burns out in 5 billion years, let N be the number of sequences that our computer could align.

5 billion years $= 5 \times 10^{12}$ years

$= 5 \times 365 \times 10^{12}$ days

$= 5 \times 365 \times 24 \times 10^{12}$ hours

$= 5 \times 365 \times 24 \times 3600 \times 10^{12}$ seconds

$\approx 15768 \times 10^{16}$ seconds

Therefore,

$(2L)^{N-2} = 10^{2N-4} = 15768 \times 10^{16}$

Taking log to the base 10 on both sides, we get:

$\log 10^{2N-4} = \log (15768 \times 10^{13})$

$\Rightarrow (2N-4) \log 10 = \log 15768 + \log 10^{13}$

$\Rightarrow 2N - 4 = \log 15768 + 13$

$\Rightarrow N = \frac{\log 15768 + 17}{2}$

$\Rightarrow N = \frac{4.197 + 17}{2}$

$\Rightarrow N = \frac{21.197}{2}$

$\Rightarrow N = \lfloor 10.098 \rfloor$

$\Rightarrow N = 10$ sequences.

Thus, in a span of 5 billion years i.e 15768×10^{16} seconds, our computer can align 10 sequences.

Ans5) The given sequences are:

$S_1 \rightarrow$ GATTCA

$S_2 \rightarrow$ GTCTGA

$S_3 \rightarrow$ GATATT

$S_4 \rightarrow$ GTCAGC

Scoring scheme: Match = 1
Mismatch/indel = −1

Now, we do pairwise alignment taking 2 sequences at a time:

For S_1 and S_2:

G A T T C A
G T C T G A

match = +3
mismatch = −3

score = match + mismatch
= 0

For S_1 and S_3:

G A TT C A
G A T A T T

match = +3
mismatch = −3
score = match + mismatch
= 0

For S_1 and S_4:

G A TT CA
G T C A GC

match = +1
mismatch = −5

score = match + mismatch
= −4

For S_2 and S_3:

G T C T G A
G A T A T T

match = +1
mismatch = −5

score = match + mismatch
= −4

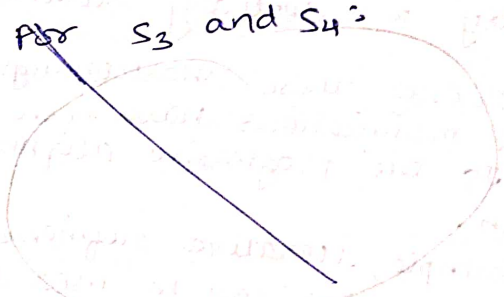
For S_2 and S_4:

G T CT G A
G T C A G C

match = +4
mismatch = −2

score = match + mismatch
= +4 −2
= +2

For S_3 and S_4:

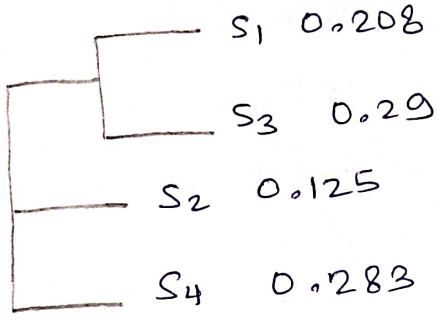


		G	T	C	T	G	A	← S_2
		G	T	C	A	G	C	← S_4
S_3 ↓	S_1 ↓							
G	G	1	−1	−1	−1	1	−1	
A	A	−1	−1	−1	0	−1	0	
T	T	−1	−1	−1	0	−1	−1	
A	T	−1	0	−1	0	−1	−1/2	
T	C	−1	0	0	−1/2	−1	−1/2	
T	A	−1	0	1	0	−1	−1/2	

The following is the score matrix:

	S_1	S_2	S_3	S_4
S_1	6	0	0	-4
S_2	0	6	-4	2
S_3	0	-4	6	-2
S_4	-4	2	-2	6

Below is phylogenetic Tree from clustal N:



S_2 and S_4 are more closely related followed by S_1 and S_3

Therefore we will align

S_2 and S_4

and,

S_1 and S_3.

Alignment:

S_1 GrATTCA
S_3 GrATATT
S_2 GrTCTGrA
S_4 GrTCAGrC

The final scores are:

$\frac{1}{4}(1+1+1+1) = 1$

$\frac{1}{4}(-1+-1+-1+-1) = -1$

$\frac{1}{4}(-1+-1+-1+-1) = -1$

$\frac{1}{4}(1+-1+-1+1) = 0$

$\frac{1}{4}(-1+-1+-1+-1) = -1$

$\frac{1}{4}(1+-1+-1+-1) = -\frac{1}{2}$

Adding up we get,
Final score $= -\frac{5}{2}$

Therefore, the final alignment is given by:

S_1: GrATTCA
S_3: GrATATT
S_2: GrTCTGrA
S_4: GrTCAGrC

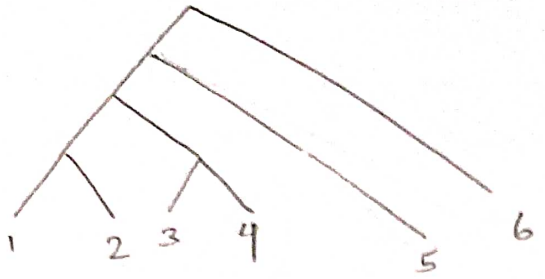
Ans 6) –

Species:	Site: 1	2	3	4
1	T	C	A	A
2	Gr	C	A	T
3	T	T	T	T
4	Gr	A	T	A
5	Gr	A	A	C
6	A	T	A	Gr

The tree given to us is:

$(((1,2),(3,4)),5),6)$

We represent the tree as:



There is
According to "Fitch-Margoliash(FM) algorithm" binary trees are used to construct phylogenetic trees based on DNA or protein sequences.

We can use this to find ancestral origin

For node 1 → {T, G}, A, C, {A, T}

Node 2 →
1st → {T, G}
2nd → {T, A}
3rd → T
4th → {T, A}

For Node 3 →
1st → {T, G}
2nd → {T, A}
3rd → {A, T}
4th → {T, A}

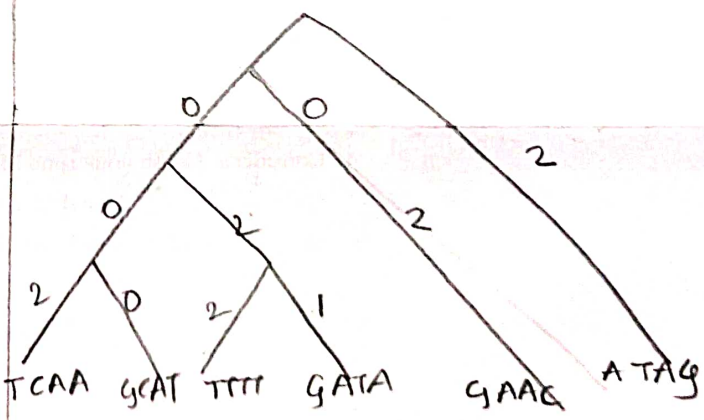
For Node 4 →
1st → {T, G}
2nd → {C, T, A}
3rd → {A, T}
4th → {G, A}

For Node 5 →
1st → {T, G, A}
2nd → {G, T, A}
3rd → {A, T}
4th → {T, A, G}

We will now choose from the above sets, such that a total substitutions are maximum

For 1st position:
G: 3 times; A: 1 time; T → 2 times
⇒ 1st position = G
2nd position : {C T A}
3rd position = A
4th position : {A, T}

We will now assign scores:



Hence,
Total ~~pari~~ parsimony score =
0 + 0 + 0 + 2 + 0 + 2 + 2 + 1 + 2 + 2
= 11

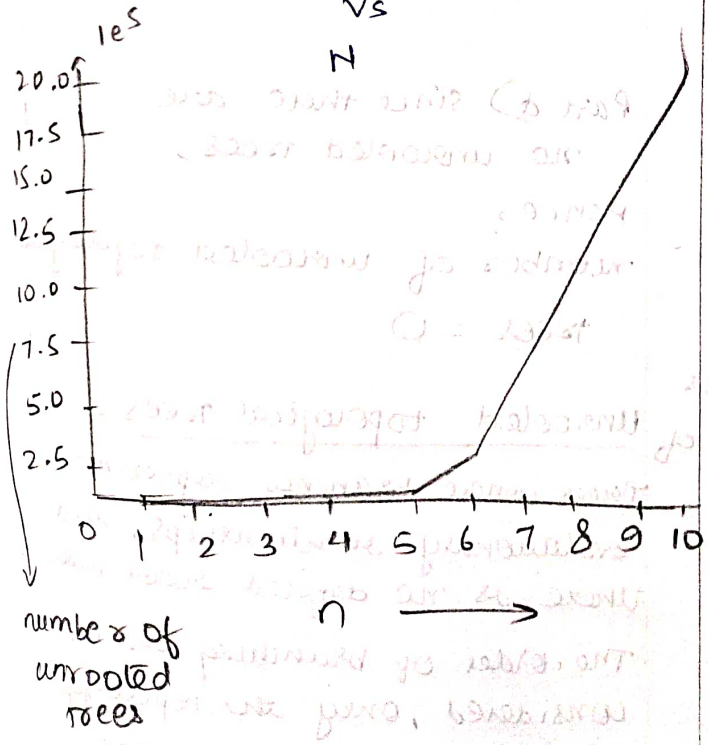
Ans 8) For n teminal taxa,

Number of unrooted trees $= \frac{(2n-5)!}{2^{n-3}(n-3)!}$

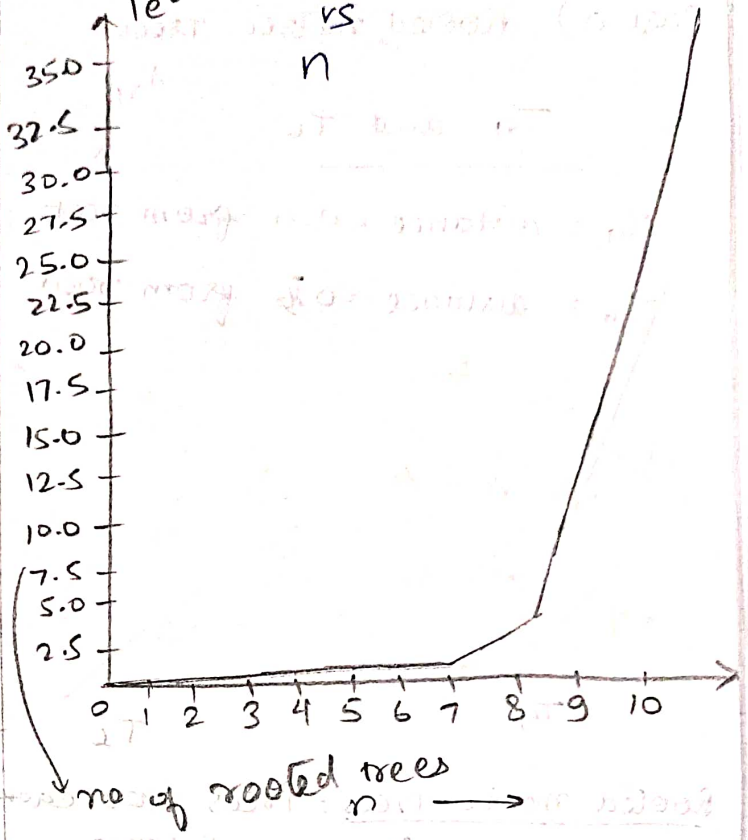
Number of Rooted trees $= \frac{(2n-3)!}{2^{n-2}(n-2)!}$

n	unrooted	Rooted
2	not defined	1
3	1	3
4	3	15
5	15	105
6	105	945
7	945	10345
8	10345	135135
9	135135	2027025
10	2027025	34459425

Number of unrooted Trees Vs N

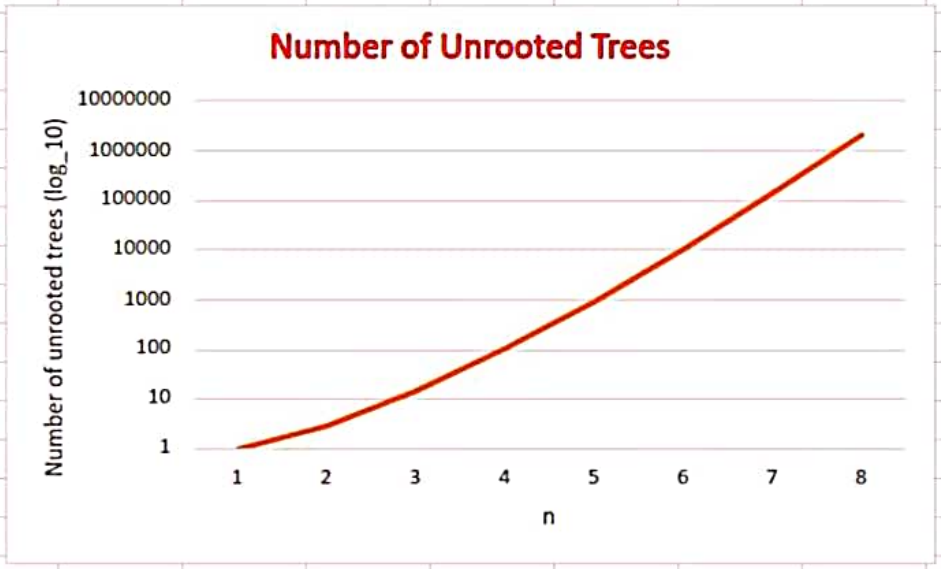


Number of Rooted Trees vs n

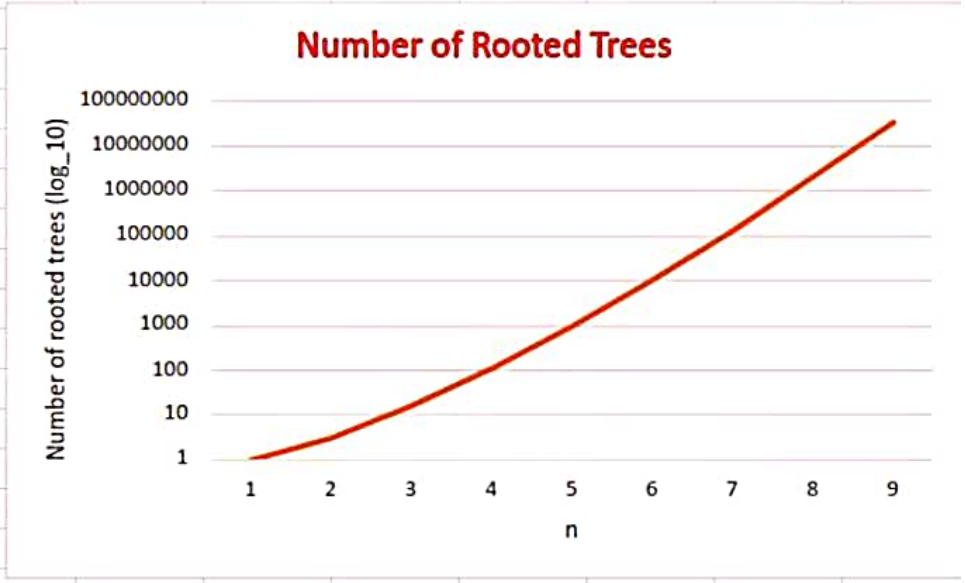


The graphs ahead are plotted with values of y-axis logarithmic to base 10.

n	Number of Unrooted Trees
3	1
4	3
5	15
6	105
7	945
8	10345
9	135135
10	2027025



n	Number of Rooted Trees
2	1
3	3
4	15
5	105
6	945
7	10345
8	135135
9	2027025
10	34459425



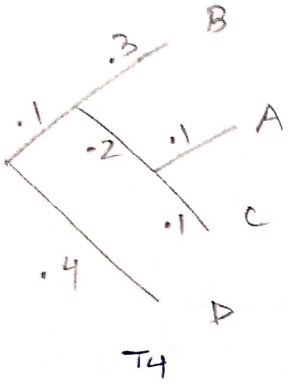
Ans 9)

Part a) Rooted metric trees

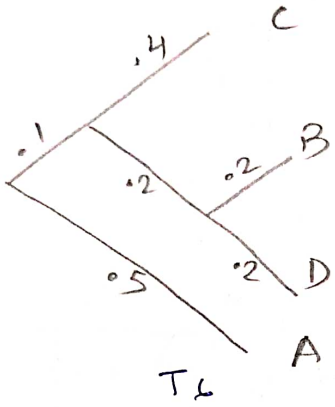
T_4 and T_6

T_4 : distance = 0.4 from root

T_6 : distance = 0.5 from root



T_4



T_6

Rooted metric trees : Trees where each branch has a length or distance, and a root node is defined as the starting point for measurements of distances between nodes.

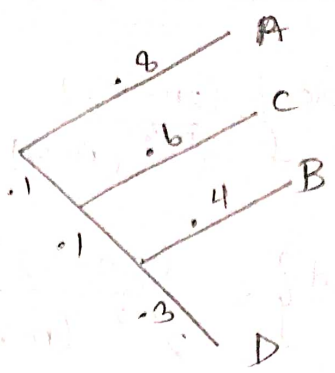
Part b) Unrooted metric trees

There is no unrooted metric tree in the figure.

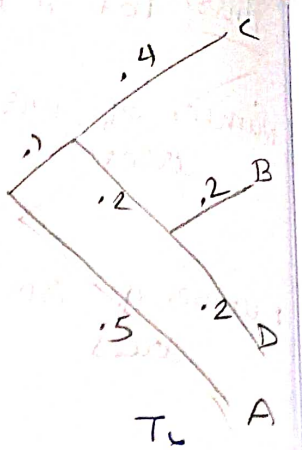
Unrooted metric trees : Trees where each branch has a length or distance, but there is no defined root node. Distances between nodes are measured by comparing pairs of nodes.

Part c) Rooted topological trees

T_1 and T_6



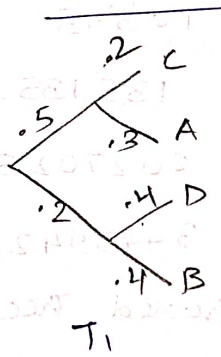
T_1



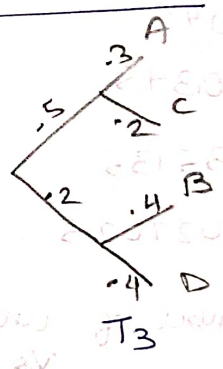
T_6

Rooted topological trees : Species which have the same ordering independent of the branch length.

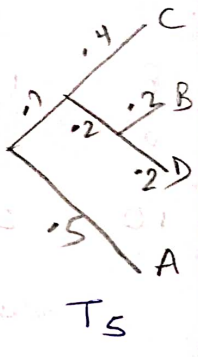
T_2 , T_3 and T_5



T_1



T_3



T_5

Part d) since there are no unrooted trees, hence, number of unrooted topological trees = 0

Unrooted topological trees : Trees where branches represent evolutionary relationships, and there is no defined root node. The order of branching is not considered, only the topology of pattern of relationships between nodes.

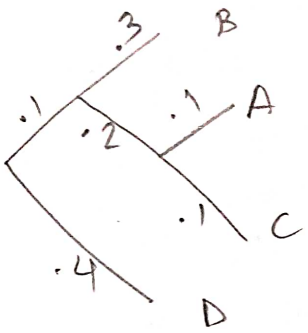
(a) part e)

A molecular clock is a concept in evolutionary biology that proposes a constant rate of molecular evolution over time, which can be used to estimate the divergence times between different species or groups.

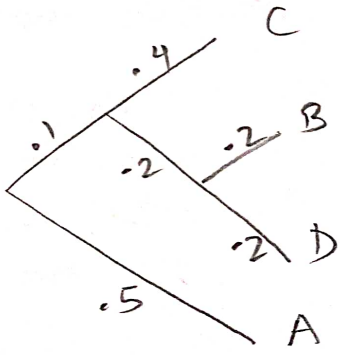
Here, trees are T4 and T6

T4 : distance = 0.4 from root

T6 : distance = 0.5 from root



T4



T6