1. Given S_1 = CCGTGTTTCA, S_2 = GTTGGCCA, S_3 = GTCCGCGCGGG, and S_4 = CGTCCCAGTTT, perform the centre star alignment computation using the following score function, s(a, b) = 0 if a = b, s(a, b) = 1 if a!= b, and s(a,−) = s(−, b) = 1. Write down the centre sequence and the intermediate alignments.

Sol.

S1: CCGTGTTTCA S2: GTTGGCCA

S3: GTCCGCGCGGG S4: CGTCCCAGTTT

Intermediate Alignments:

Centre Sequence Alignment:

S1: - - - CCGTGTTTCA

S2: - - - GTTGG- - CCA

S3: -GTCCGCG- CGGG

S4: CGTCCCAGTTT - -

*

$\sum_{i=1..k} D(S1, Si) = 21$

$\sum_{i=1..k} D(S2, Si) = 21$

$\sum_{i=1..k} D(S3, Si) = 21$

$\sum_{i=1..k} D(S4, Si) = 21$

	S_1	S_2	S_3	S_4
S_1	0	6	8	7
S_2		0	7	8
S_3			0	6
S_4				0

2. Given four species {A,B,C,D} with a distance defined as: d(A,B) = 10, d(A,C) = 13, d(A,D) = 18, d(B,C) = 6, d(B,D) = 13, d(C,D) = 12.

Use the neighbour-joining algorithm to produce the evolutionary tree of {A, B, C, D}.

Sol. Distance Matrix:

	A	B	C	D
A	0	10	13	18
B		0	6	13
C			0	12
D				0

i	u_i
A	(10 + 13 + 18) / 2 = 20.5
B	(10 + 6 + 13) / 2 = 14.5
C	(13 + 6 + 12) / 2 = 15.5
D	(18 + 13 + 12) / 2 = 21.5

	A	B	C	D
A	0	-25	-23	-24
B		0	-24	-23
C			0	-25
D				0

9.25 D

3 C

Further Steps:

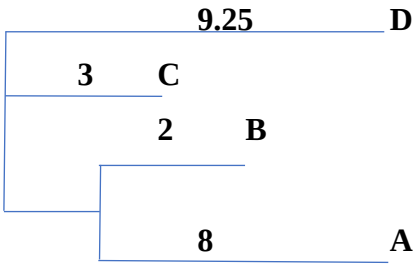
	A	B	C	D	X
A	0	10	13	18	9.5
B		0	6	13	3.5
C			0	12	
D				0	
X					0

	A	B	X
A	0	10	9.5
B		0	3.5
X			0

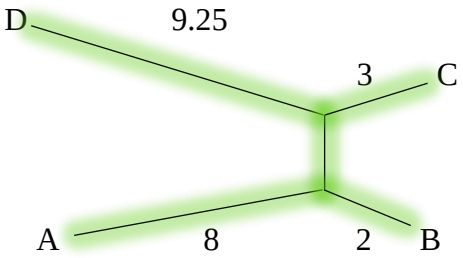
i	u_i
A	19.5
B	13.5
X	13

	A	B	X
A	0	-23	-23
B		0	-23
X			0

Phenogram:



Resulting Phylogenetic Tree:



3. Let T be a labelled binary phylogenetic tree with n ≥ 3 leaves, prove that there are $\prod_{j=3}^{n}(2j-5)$ different trees.

Sol.

Given the number of possible leaf-labelled binary trees, where each leaf has a different label drawn from {1, 2, ..., n}, is relevant to the approximation of phylogenetic trees.

n	Total Binary Trees
4	3
5	15
6	105
7	954
8	10395
9	135135
10	2027025

We know, for n = 1, 2, 3, only 1 phylogenetic tree is possible. Therefore, when n ≤ 3, there is only one unrooted binary tree. Whereas, when n = 4, there are 3 possible trees that could be constructed.

Furthermore, this can be seen algorithmically:

To construct a tree with n = 4 leaves (i.e., with leaves S1, S2, S3, S4).

Let us consider, a tree with three leaves (n = 3) and then add remaining leaf by subdividing an edge in tree T and make S4 next to the newly introduced node. Accordingly, the number of possible trees with n=4 leaves is equivalent to the number of edges in T. Since T has three leaves, it has exactly three edges.

Hence, there are three unrooted binary trees with 4 leaves.

Similar algorithmic analysis can be applied to generalise the above condition for phylogenetic trees with n ≥ 3 leaves.

Let us consider, a tree T with (n - 1) leaves, pick an edge in tree T and subdivide it and make S_n next to the newly created node. The number of unrooted binary trees on n leaves is consequently equivalent to the product of the number T_{n-1} of unrooted binary trees on n - 1 leaves and the number e_{n-1} of edges in an unrooted tree n - 1 leaves.

Therefore, we get the number of unrooted trees:

$t_n = (2(n-1)-3)\, t_{n-1} = (2n-5)\, t_{n-1} = (2n-5) * (2n-7) * \ldots * 3 * 1 = (2n-5)! / ((n-3)!2^{n-3})$, $n > 2$

Number of rooted trees Tn, $Tn = (2n-3)\, t_n = (2n-3)! / ((n-2)!2^{n-2})$, $n > 2$; i.e., the number of rooted trees times the number of branches in the trees.

Hence using maximum likelihood formula to generalize the estimation,

there are $\prod_{j=3}^{n}(2j-5)$ different trees, for a labelled binary phylogenetic tree with n ≥ 3 leaves.

4. In UPGMA clustering algorithm, the distance between two clusters C_i and C_j can be defined as the average distance between pair of objects from each cluster:

$d_{ij} = \frac{1}{|C_i||C_j|} \sum_{p \epsilon C_i, q \epsilon C_j} \boldsymbol{d_{pq}}$

Prove that if C_k is the union of C_i and C_j and if C_l is any other cluster, then

$$d_{kl} = \frac{d_{il}|C_i| + d_{jl}|C_j|}{|C_i| + |C_j|}$$

Sol. UPGMA (Unweighted Pair Group Method)

Basically, we are given a matrix of pairwise distances between each pair of sequences, which starts with assigning each sequence to its own cluster.

The distances between the clusters are defined as

$$d_{ij} = \frac{1}{|C_i||C_j|} \sum_{p \epsilon C_i, q \epsilon C_j} \boldsymbol{d_{pq}} \qquad \textbf{(4.1)}$$

Where C_i and C_j denotes sequences in clusters i and j, respectively. At each stage in the process, the least distant pair of clusters are merged to create a new cluster. This process continues until only one cluster is left. We will perform n-1 iterations.

In equation 4.1, two identical clusters C_i and C_j are linked with a new root to form a big cluster.

So, set C = {{c_1}, {c_2}, …, {c_n}} and height ({c_i}) = 0 for i ∈ {1, …., n}

For all {c_i}, {c_j} ∈ C, set d_{ij} = M_{ij}

For i = 2 to n do:

Determine clusters C_i, C_j ∈ C such that d_{ij} is minimized.

Let C_k be a cluster formed by connecting C_i and C_j to the same root.

Let height (C_k) = d_{ij} /2

Suppose d_{ki} = height(C_k) – height(C_i) and d_{kj} = height(C_k) – height(C_j)

C = C – {C_i, C_j} ∪ {C_k}

Therefore, for all C_l ϵ C – {C_k}, declare

$$d_{Cl} = \frac{\sum_{i \epsilon C, j \epsilon C_l} \boldsymbol{M_{ij}}}{|C||C_l|} = \frac{\sum_{i \epsilon C1, j \epsilon C_l} \boldsymbol{M_{ij}} + \sum_{i \epsilon C2, j \epsilon C_l} \boldsymbol{M_{ij}}}{|C||C_l|}$$

So, $d_{il} = \frac{\sum_{i \epsilon C, j \epsilon C_l} \boldsymbol{M_{ij}}}{|C_i||C_l|}$ and $d_{jl} = \frac{\sum_{i \epsilon C, j \epsilon C_l} \boldsymbol{M_{ij}}}{|C_j||C_l|}$

Therefore, for all C_l ∈ C – {C_k},

Hence proved, $d_{kl} = \frac{d_{il}|C_i| + d_{jl}|C_j|}{|C_i| + |C_j|}$

5. The consensus model of multiple alignment is a special case of tree alignment. Simplify the 2-approximation tree alignment algorithm proof to show a 2-approximation consensus multiple alignment algorithm proof.

Sol.

We know, in Tree alignment, the score is the sum of the pairwise score on each edge of the tree as in Consensus sequence method, we measure the score of the multiple sequence alignment with the sum of pairwise differences. Accordingly, with respect to the consensus model the multiple alignment produced by this algorithm is also an approximate solution.

Primarily, for calculating the optimal multiple alignment under the SP metric we present an approximation algorithm (with an approximation ratio of 2).

Let us consider, the centre star, T_c to be a star tree of k nodes, with the centre node labelled S_c and with each of the $k-1$ remaining node labelled by a distinct sequence in $S / \{S_c\}$.

The multiple alignment M_c of S is the multiple alignment induced by the centre star,

i.e., for each $v <> c$, the alignment Mc induces an optimal pairwise alignment between S_c and S_v.

Let W to $be \sum_{i \neq c} D(S_c, S_j)$ and we get:

$$2d\,(M_c) \leq 2\,(k-1)\,W$$

On the other hand, by choice of c it follows that:

$$2d\,(M*) = \sum_{i \neq j} d * (S_i, S_j) \geq \sum_{i \neq j} D(S_i, S_j)$$

$$= \sum_i \sum_{j \neq i} D(S_i, S_j) \geq \sum_i W = kW$$

And finally,

$$\frac{d(MC)}{d(M*)} \leq \frac{2(k-1)W}{kW} = \frac{2(k-1)}{k}$$

2 approximation consensus multiple alignment algorithm:

For $\bar{S} \in S$:

$$E(\bar{S}) = \sum_{S_i \in S} D(\bar{S}, S_i) \leq \sum_{S_i \neq \bar{S}} [D(\bar{S}, S*) + D(S*, S_i)]$$

$$(k-2) \cdot D(\bar{S}, S*) + D(\bar{S}_i S*) + \sum_{S_i \neq \bar{S}} D(S*, Si) = (k-2) \cdot D(\bar{S} i S*) + E(S*)$$

If we pick $\bar{S} \in S$ pick such that $\bar{S}$ is closest to $S*$ then:

$$E(S*\bar{}) = \sum_{S_i \in S} D(S*, S_i) \geq k \cdot D(\bar{S}, S*)$$

The centre string $S_c \in S$ minimizes $\sum_{i \neq c} D(Sc, Sj)$ and therefore its consensus error is smaller than the consensus error of the $\bar{S}$ ($the\ string\ closest\ to\ S*$).

We get,

$$\frac{E(Sc)}{E(S*)} \leq \frac{(k-2) \cdot D(Sc,S*) + E(S*)}{E(S*)} \leq \frac{(k-2) \cdot D(Sc,S*) + E(S*)}{k \cdot D(Sc,S*)} + 1 = 2 - \frac{2}{k} = \frac{2(k-1)}{k}$$

Therefore, we simplify the 2-approximation tree alignment algorithm proof to show a 2-approximation consensus multiple alignment algorithm proof.