ASSIGNMENT 3 DIFFICULT - SOLUTION - 180532 & 180653

Algorithm:

Consider the graph **G** with nodes as all the points in the set **P**. All the nodes are connected to each of the other nodes by an edge. Any edge between p_i and p_j is of the length/value $d(p_i, p_j)$, where d is the distance function defined in the question. To define the τ function, we construct a Minimum Spanning Tree **T** from the graph **G** in the following manner:

- Start with the edge with the lowest value.

- Add both the nodes of this edge to the set **C** and to our tree as the leaf nodes of the tree, and set the node value of their parent as the edge value, i.e.

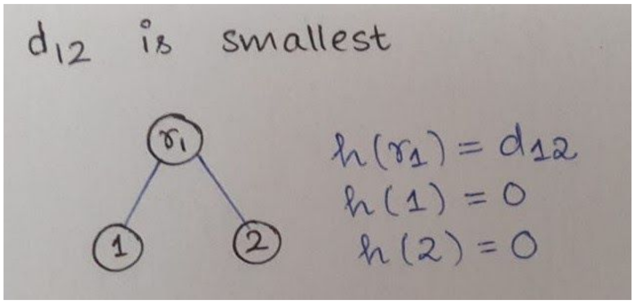
  h(parent of p_i *and* p_j*)* = $d(p_i, p_j)$, where i and j form the smallest edge.

- Now look for the next smallest edge in the graph such that one node of the edge is in the set **C** and the other node is not in **C**, so that we avoid formation of cycles. Let this new exterior node be p_k, then add p_k to the tree as a leaf node. Introduce a parent node for p_k and our previously defined tree with node value as edge length.

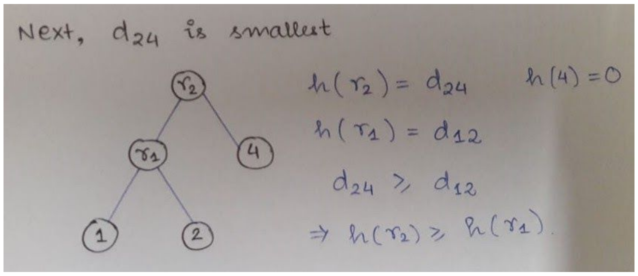
- We repeat the third step till we have included all the nodes, i.e. all the points in the tree.

At any moment, we have a set C consisting of the already included nodes and a set of excluded nodes. We pick the node from the excluded set which has the minimum length edge to any node in C and add it to the tree. The new tree is made by including a parent with h(parent) as the edge length and its two children are the new node as a leaf node and the old tree.

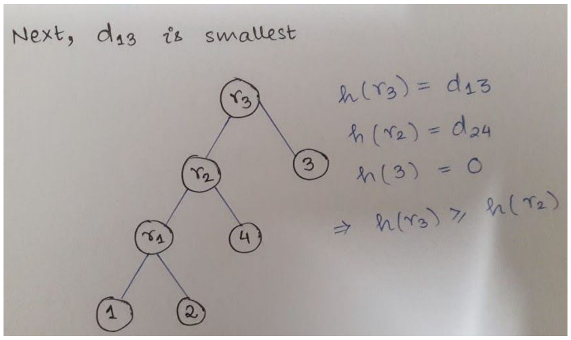
Example: Consider the set P with four nodes. We start creating the MST using our algorithm.



Here, d_{12} is the smallest edge and we include 1st and 2nd nodes in the tree. We also introduce a parent node r_1.



Here, d_{24} is the next smallest edge. We introduce a parent node r_2 which has children r_1 and the 4th node. Here we can see that $d_{14} <= d_{24}$, because d_{24} is known to be the next smallest edge. So, $h(r_2) <= d(i, j)$ for all i and j whose lowest common ancestor is r_2.



Here, the set C contains node 1, 2 and 4 already and we are left with node 3 only. Of all the edges from node 3, d_{13} is the smallest so $h(r_3) = d_{13}$. Again, because d_{13} is the smallest edge, all the edge values from any node in the subtree of r_2 to node 3 must be greater than or equal to $h(r_3)$, i.e. d_{13}.

Proof Of Correctness:

Consider the tree **T**, constructed using the metric τ. In this tree, we are including the nodes(points in the set) one by one based on their distances. Thus, when we include a new node p_j, we have a set **C** of nodes(or points) that have already been included. Let **x** be the point nearest to p_j such that **x** belongs to C.

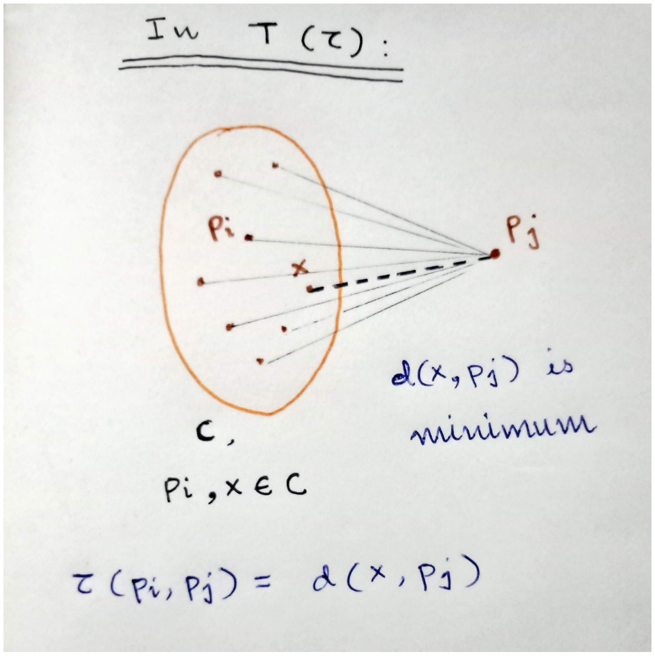
Thus, we have $h(LCA(i,j)) = d(\mathbf{x},p_j) = \tau(p_i, p_j)$

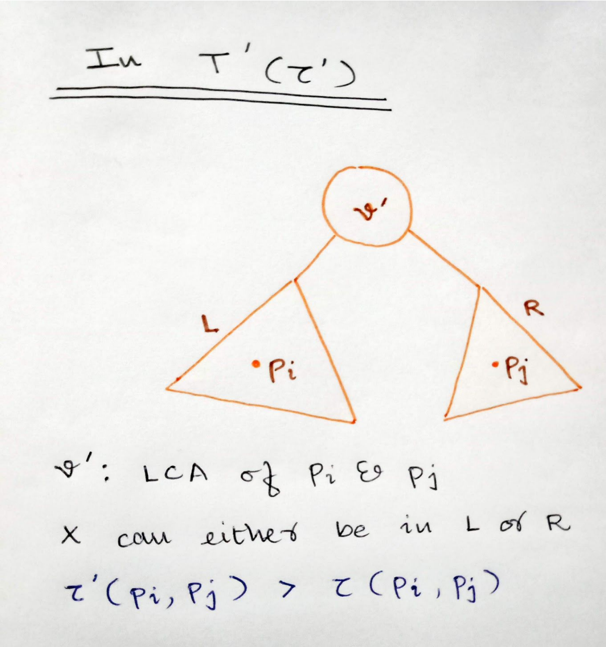
Now, we have to satisfy the following conditions:

- If x is parent of y, $h(x) \geq h(y)$.
  For any leaf node p_n , $h(p_n) = 0$.
  We are always choosing edges in increasing order of their distances. Thus, when we create a new node to connect tree with set C and a point outside C, we take a new edge with one point $p \in C$ and another point x, outside C, such that the edge distance is least possible. All other edges in the MST had distance less than d(p, x). Thus, all the nodes in the subtree containing p have height value less than d(p, x). This subtree is connected with x through a node v with h(v) = d(p, x) and h(x) = 0. Thus, it is maintained at each point of tree construction that the height of a node is always greater than or equal to that of its children.

- τ is consistent with d: $\tau(i,j) \leq d(i,j)$
  It is clear that when we include a new point in the tree T, we add a new root node between the existing tree(containing set C) and the new node p_j with $h(root) = d(\mathbf{x}, p_j)$. Since $d(\mathbf{x}, p_j)$ is the least among all distances from points in C to j, therefore $\forall$ points $p \in C$, we have:
  $d(p, p_j) \geq d(\mathbf{x}, p_j) = \tau(p,p_j)\ \forall$ p. Thus, this condition is always satisfied.

- For any other hierarchical metric τ' consistent with d, we must have $\tau'(p_i, p_j) \leq \tau(p_i, p_j)$ for each pair of points p_i and p_j.
  We prove that there exists no such metric τ' such that $\tau'(p_i, p_j) > \tau(p_i, p_j)$ using contradiction. Let us assume there exists a metric τ' such that $\tau'(p_i, p_j) > \tau(p_i, p_j)$ for some pair of points p_i & p_j.
  We have already defined the point **x** in **T** (with metric τ). Let LCA(p_i, p_j) = **v** in **T**.
  $h(\mathbf{v}) = d(\mathbf{x},p_j) = \tau(p_i, p_j)$
  For metric τ', we have tree **T'** and let the LCA(p_i, p_j) = **v'**.





The node **x** can either be in the right subtree of **v'** or the left subtree.

- <u>When x is in the left subtree of v'</u>:

  $\tau(p_i, p_j) = d(\mathbf{x}, p_j)$
  But, $\tau'(p_i, p_j) > \tau(p_i, p_j)$
  => $\tau'(p_i, p_j) = \mathbf{h(v') > d(x, p_j)}$.

  Since τ' is consistent with d, d(**x**, p_j) should be satisfied too.
  This gives: $d(\mathbf{x}, p_j) \geq \tau'(\mathbf{x}, p_j)$
  As LCA(**x**, p_j) = LCA(p_i, p_j) = **v'**
  => $\tau'(\mathbf{x}, p_j)$ = h(**v'**)
  Thus, we have: ***d(x, p_j) ≥ h(v').***
  The two statements are in **contradiction** with each other. Thus, our assumption was wrong. If **x** is in the left subtree, we cannot have $\tau'(p_i, p_j) > \tau(p_i, p_j)$ for any pair of points p_i & p_j.

- <u>When x is in the right subtree of v'</u>:

  The tree T is formed using the MST construction method. Since x belongs to set C and was included before point pj was included, it follows that, for all edges with points p,q included in the MST, ***d(p, q) ≤ d(x, p_j)***, as edge(p,q) was added before edge (x, pj) during MST construction.

We prove that for any edge in the MST of T, all the points should belong to the same subtree of v' in T' as well.

For T', we know: $\tau'(p_i, p_j) > \tau(p_i, p_j)$ => $\boldsymbol{h(v') > d(x, p_j)}$.

For any edge in the MST of T with end vertices p, q we have:
$d(p, q) \leq d(x, p_j)$ => $\boldsymbol{d(p, q) \leq h(v')}$.
If p, q are in different subtrees of v', LCA(p, q) = v'.
Thus, $\boldsymbol{\tau'(p, q) = h(v') > d(p, q)}$

This **contradicts** the property of τ' that it is consistent with d. Thus, our assumption was wrong. We must have all the points in the set C in the same subtree. If **x** is in the right subtree, we cannot have $\tau'(p_i, p_j) > \tau(p_i, p_j)$ for any pair of points p_i & p_j.

The above two cases show that we cannot have any hierarchical metric τ', such that τ' consistent with d and $\tau'(p_i, p_j) > \tau(p_i, p_j)$ for any pair of points p_i & p_j.

Time Complexity

We are using the MST construction method for this algorithm. We know that there exist many algorithms with polynomial time complexity for creating a MST. Other steps in the algorithm like creating a new node and updating its height value are O(1) time step. Thus, overall time complexity is bounded by a polynomial function.