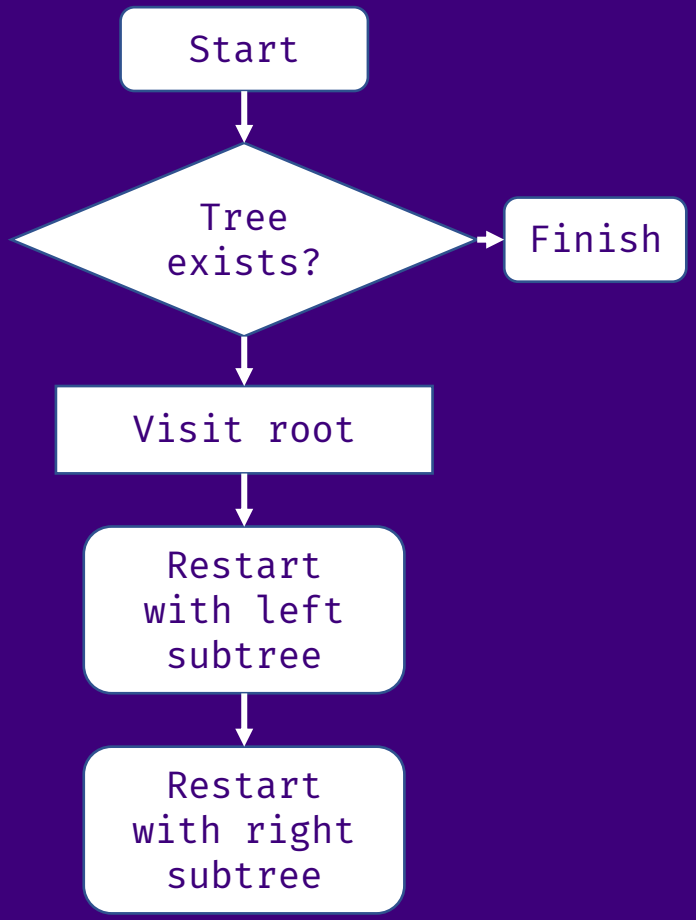
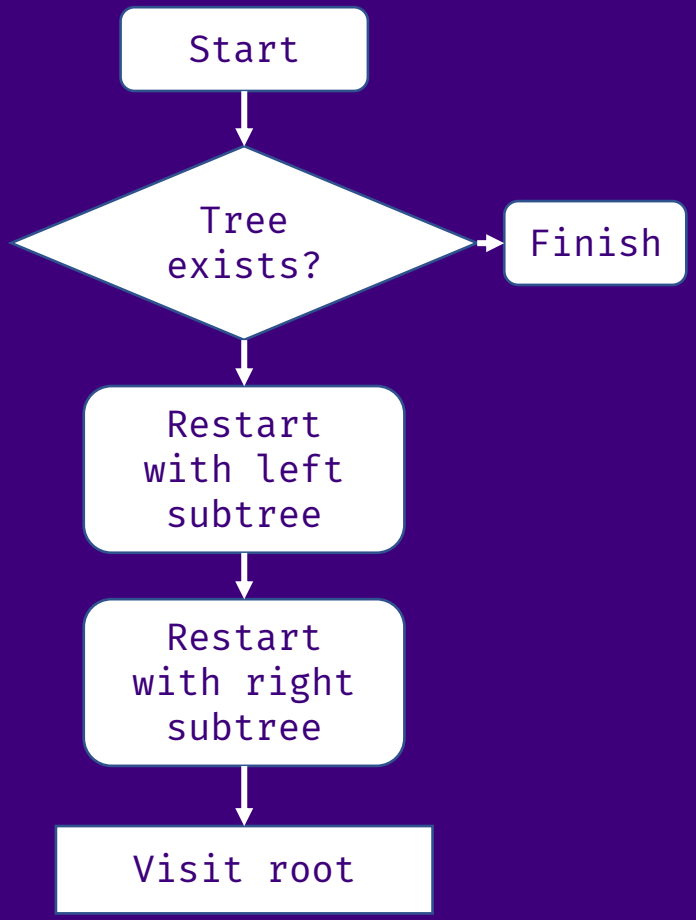
Brownbag: Algorithms & Problem Solving with Graphs and Trees

Traversing a binary tree

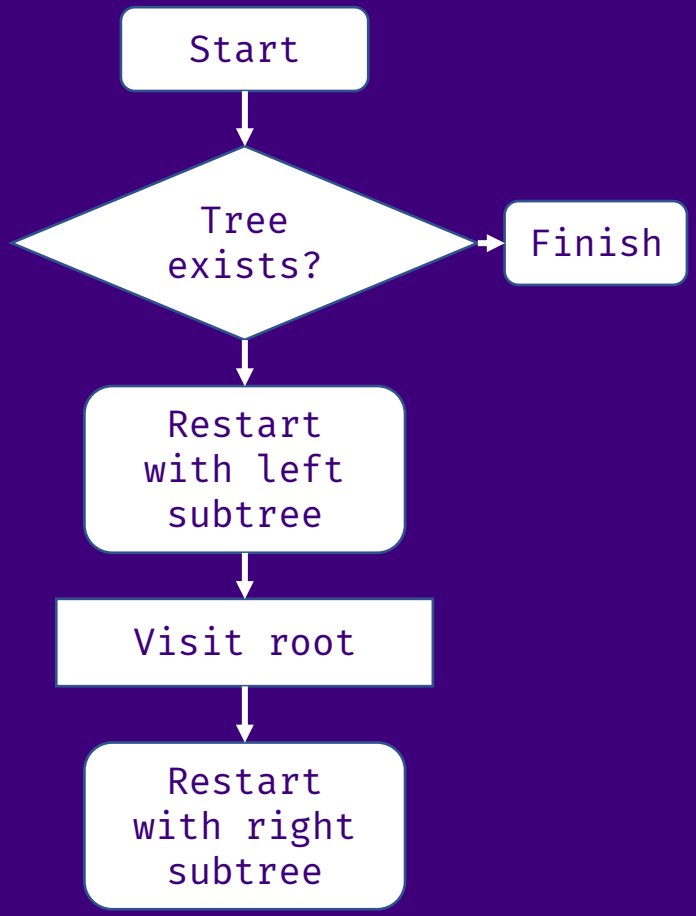
Preorder algorithm



Postorder algorithm

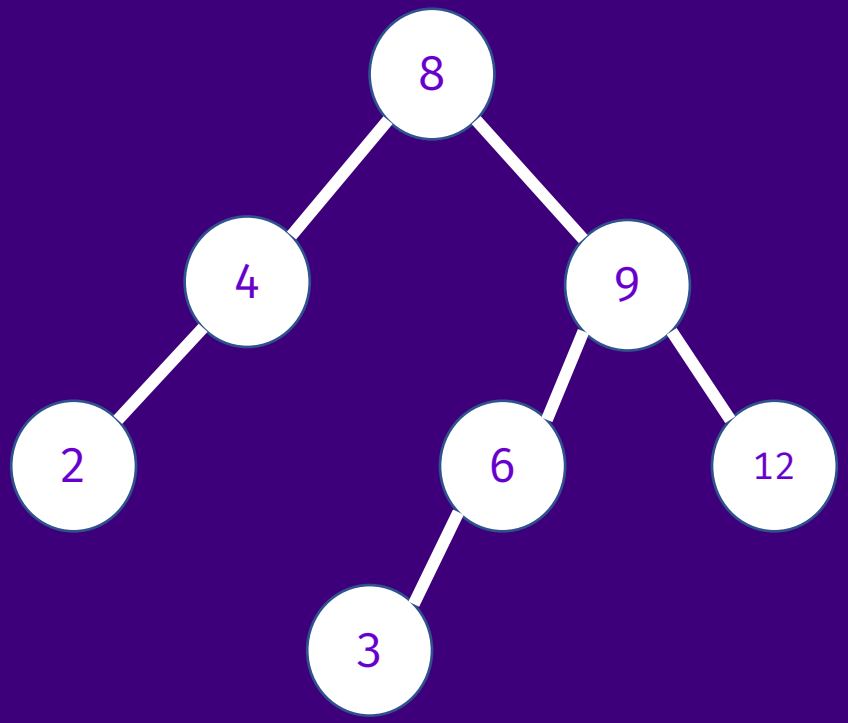


Inorder algorithm



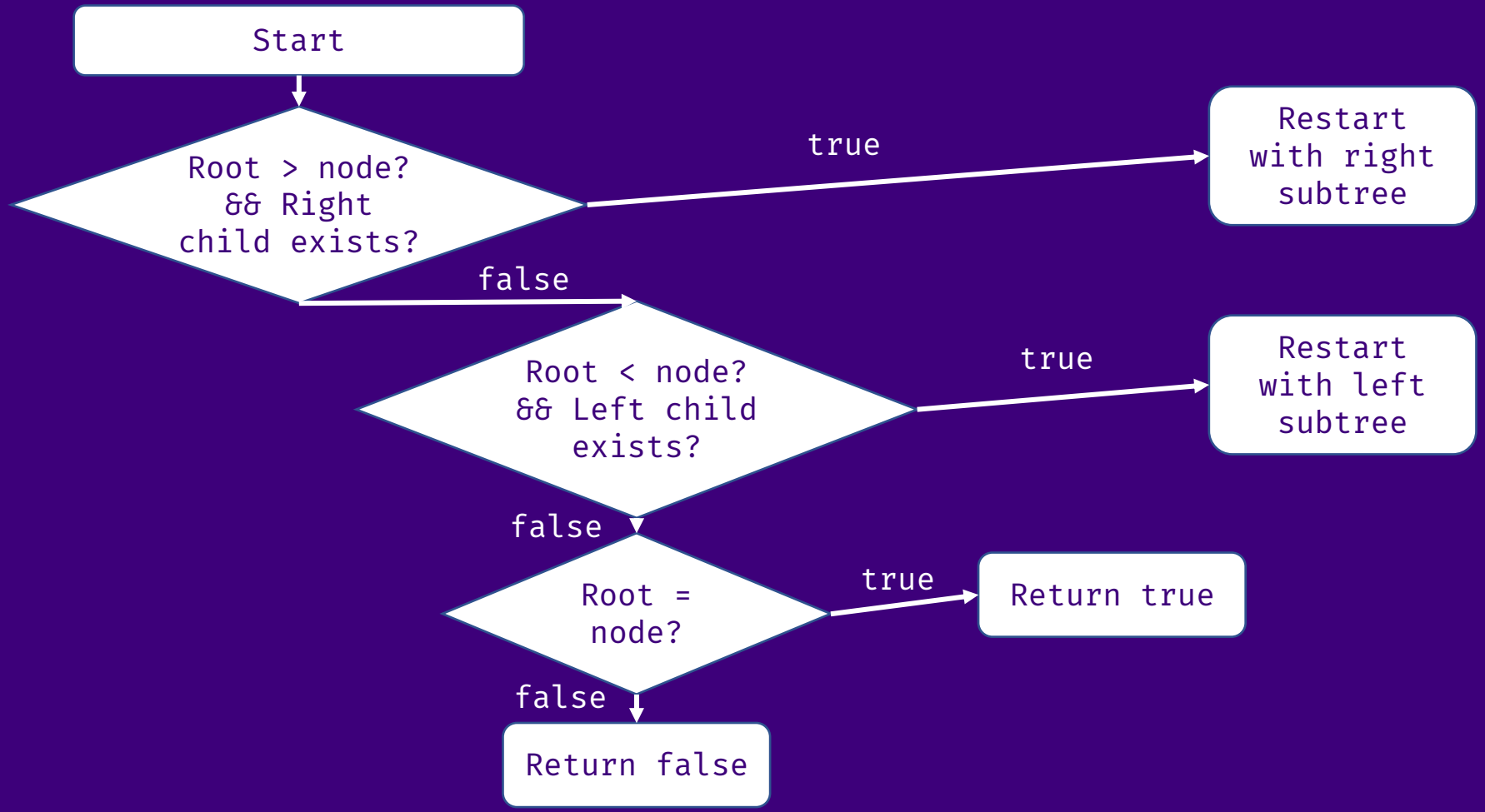
Binary search tree

Binary search tree property: For each node, all keys in the left subtree are less than the node, and all keys in the right subtree are greater than the node.



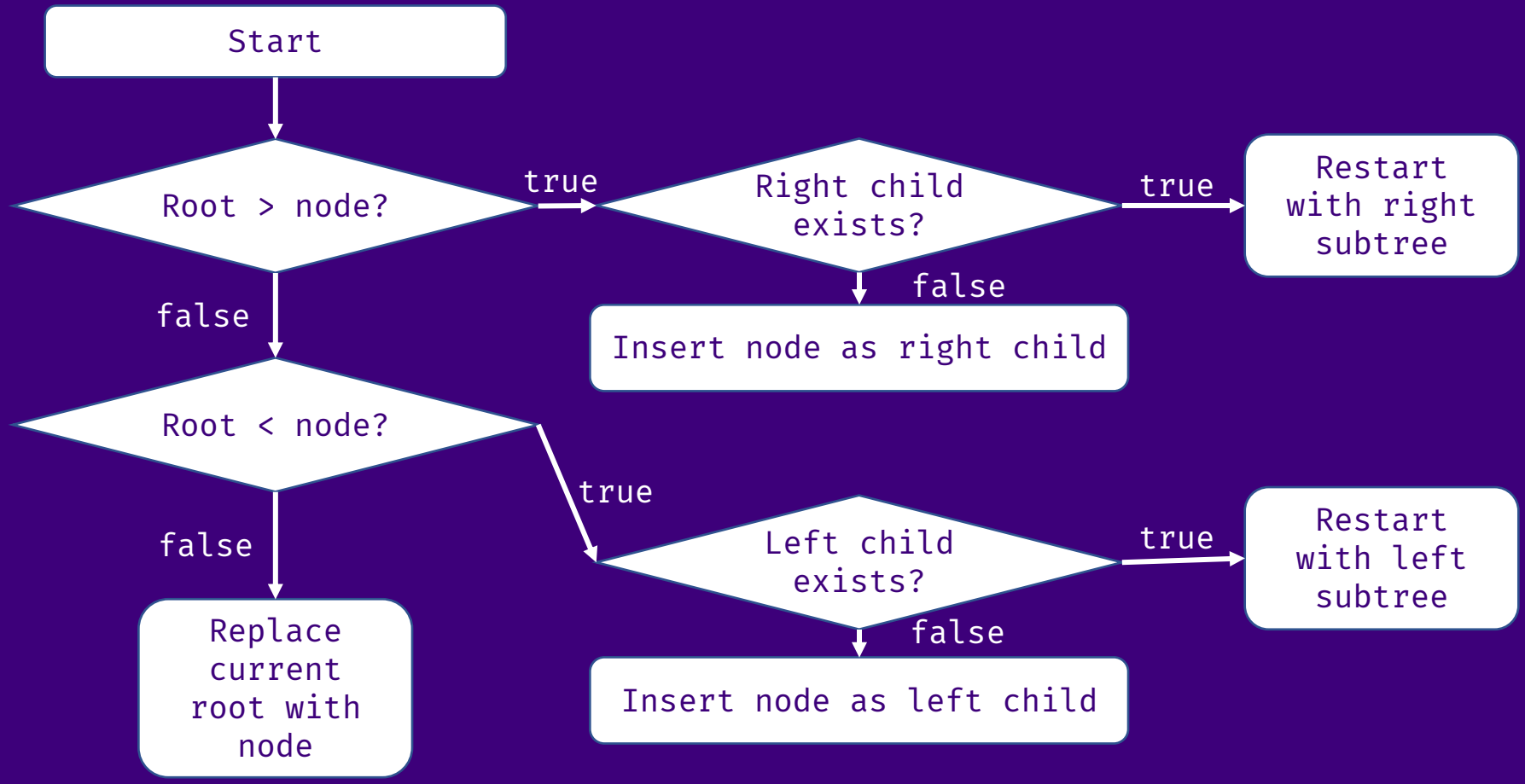
Binary search tree

Traversal algorithm



Binary search tree

Insertion algorithm



Prim’s algorithm

- Used to find a minimum spanning tree (MST) of a graph.
- A greedy algorithm: At each step, the algorithm will choose the ‘cheapest’ next step.

Dijkstra's algorithm

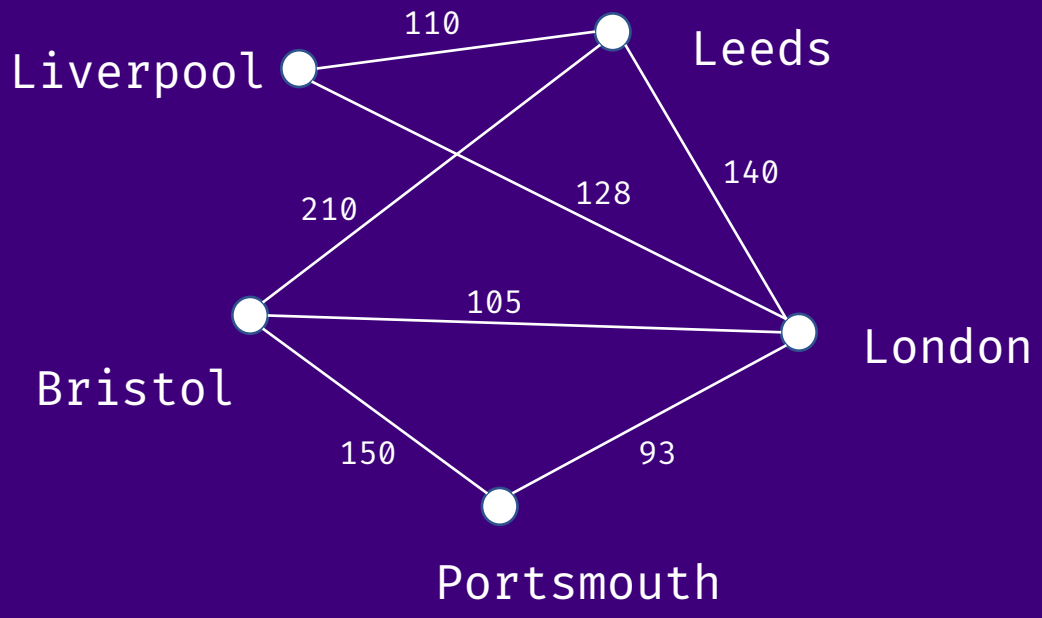
- Used to find the shortest path between two nodes in a graph.
- Another greedy algorithm.

Traveling salesperson problem

- What is the shortest route through the graph that visits all the nodes:
  - Only once (classical problem)
  - At least once (practical problem)

Train travel times: English cities

Visualization

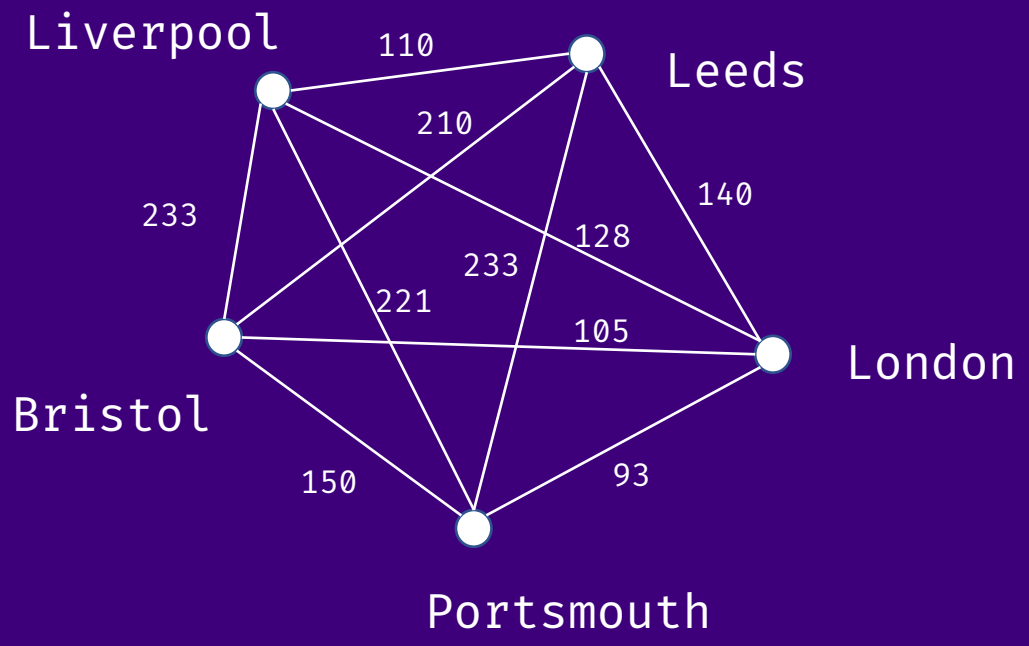


Adjacency list

Node	
1	Id = “Leeds” Adj = {2:110, 3:140, 4:210}
2	Id = “Liverpool” Adj = {1:110, 3:128}
3	Id = “London” Adj = {1:140, 2:128, 4:105, 5:93}
4	Id = “Bristol” Adj = {1:210, 3:105, 5:150}
5	Id = “Portsmouth” Adj = {3:93, 4:150}

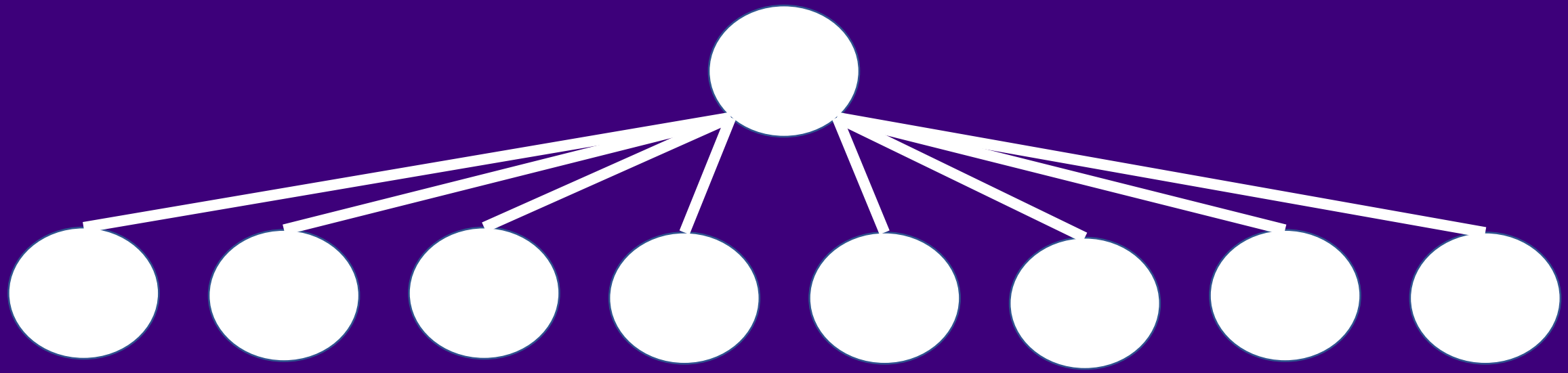
Train travel times: English cities

Complete network of least distances



Keys	Liverpool	Leeds	Bristol	London	Portsmouth
Liverpool	-1	110	233	128	221
Leeds	110	-1	210	140	233
Bristol	233	210	-1	105	150
London	128	140	105	-1	93
Portsmouth	221	233	150	93	-1

Octree



Parse tree

Virtual Coffee is building an awesome community.