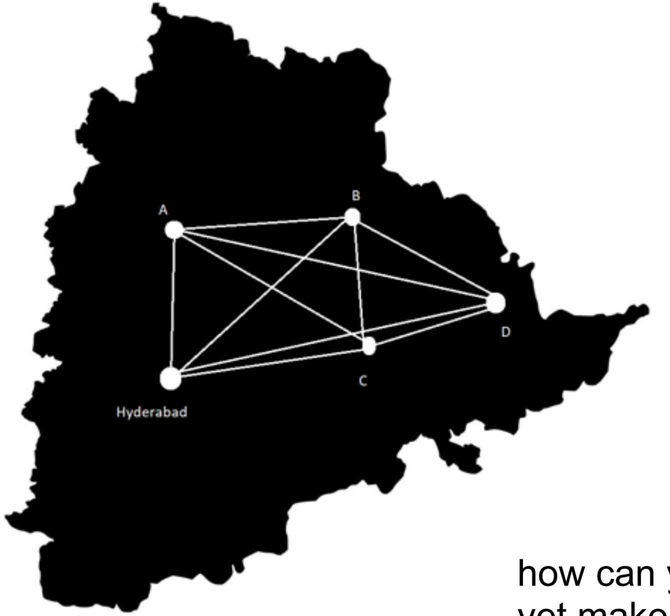
Design and Analysis of Algorithm

Lecture-19:
Dynamic Programming

Contents

Travel

Suppose you decide to drive car around Telangana and you start your journey from Hyderabad



	Hyderabad	A	B	C	D
Hyderabad	–				
A	425	–			
B	167	257	–		
C	306	209	219	–	
D	323	105	198	105	–

how can you minimize the number of kilometers
yet make sure you visit all the cities?

Tour

Since there are only five cities it's not too hard to figure out the optimal tour

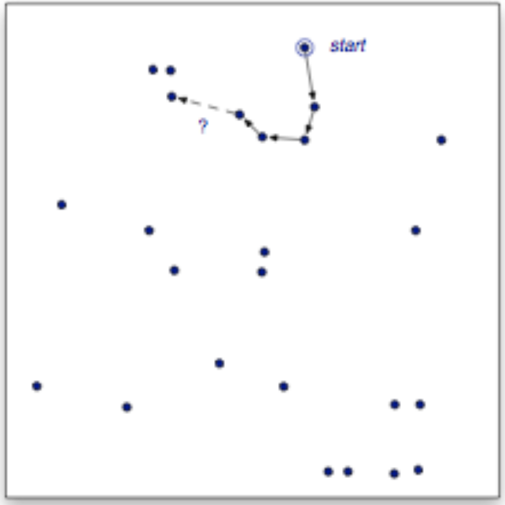
One way to find the optimal tour is to consider all possible paths

There is a problem with the exhaustive search strategy

- the number of possible tours of a map with n cities is $(n - 1)! / 2$

#cities	#tours
5	12
6	60
7	360
8	2,520
9	20,160
10	181,440

As we add cities to our tour, however, it is much harder to figure out the optimal tour

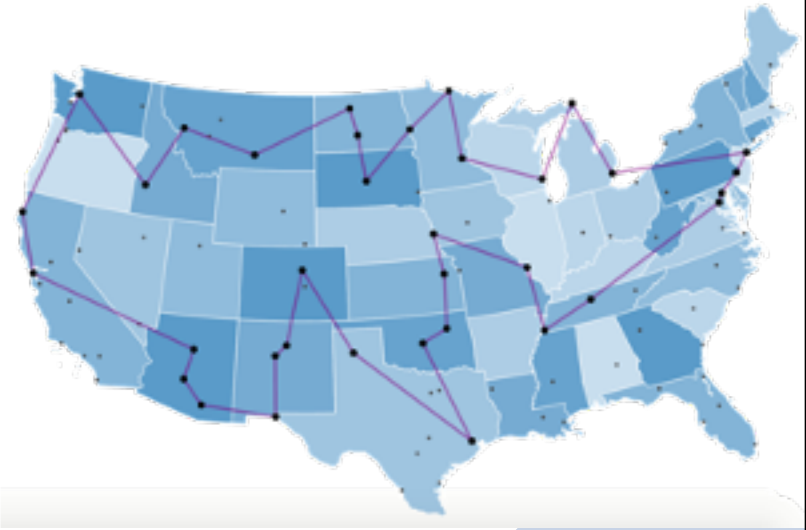


Problem Statement

Definition

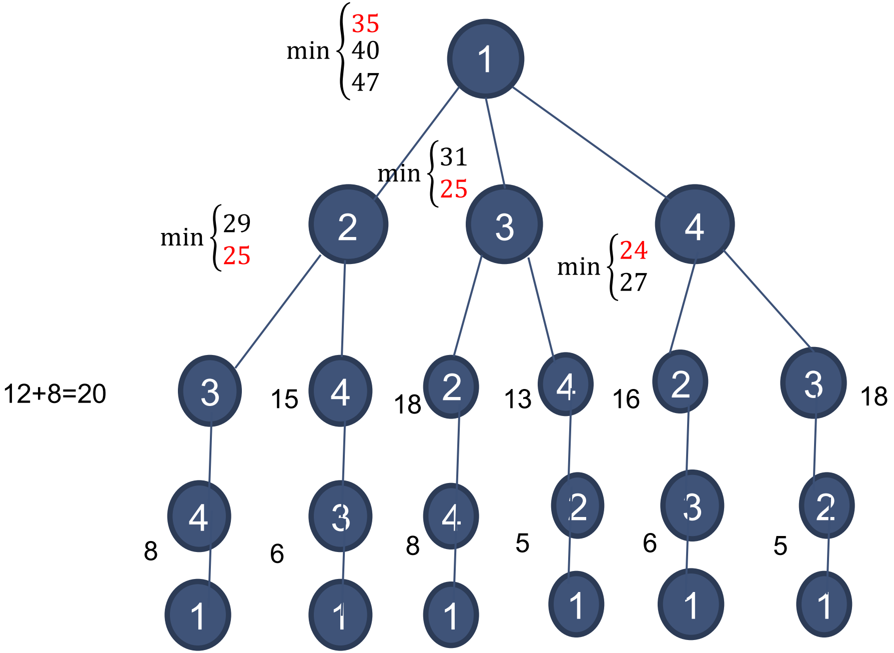
A traveler needs to visit all the cities from a list, where distances between all the cities are known and each city should be visited just once.

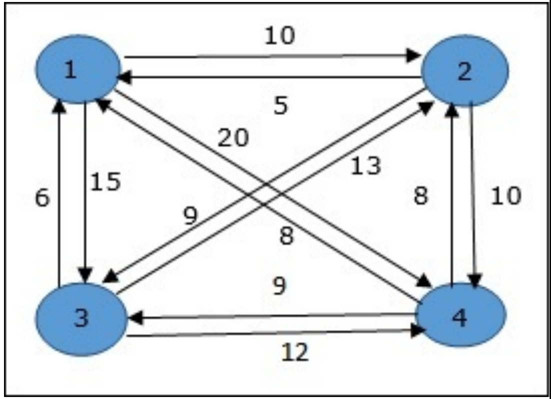
What is the shortest possible route that he visits each city exactly once and returns to the origin city?



Example

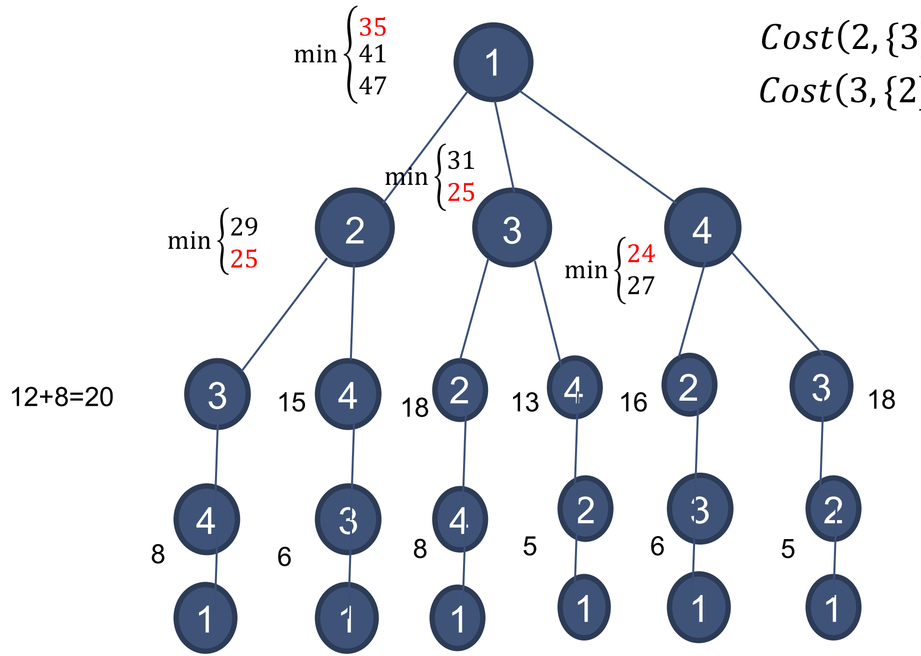
The cost to visit from one to another city is given in the graph. Find the optimal route to visit from node 1 to all other once node and then come back to 1





Mathematical Steps

$$Cost(i,S) = \min_{j \in S}\{c_{ij} + Cost(j, S - \{J\})\}$$



$Cost(2,\Phi) = 5,\quad Cost(3,\Phi) = 6,\quad Cost(4,\Phi) = 8,$

$Cost(3,\{4\}) = c_{34} + cost(4,\Phi) = 20$

$Cost(4,\{3\}) = c_{43} + cost(3,\Phi) = 15$

$Cost(2,\{4\}) = c_{24} + cost(4,\Phi) = 18$

$Cost(4,\{2\}) = c_{42} + cost(2,\Phi) = 13$

$Cost(2,\{3\}) = c_{23} + cost(3,\Phi) = 16$

$Cost(3,\{2\}) = c_{32} + cost(2,\Phi) = 18$

$$Cost(2,\{3,4\}) = \min\begin{cases} c_{23} + cost(3,4) = 9 + 20 \\ c_{24} + cost(4,3) = 10 + 15 \end{cases}$$

$$Cost(3,\{2,4\}) = \min\begin{cases} c_{32} + cost(2,4) = 13 + 18 \\ c_{34} + cost(4,2) = 12 + 13 \end{cases}$$

$$Cost(4,\{2,3\}) = \min\begin{cases} c_{42} + cost(2,3) = 8 + 16 \\ c_{43} + cost(3,2) = 9 + 18 \end{cases}$$

$$Cost(1,\{2,3,4\}) = \min\begin{cases} c_{12} + cost(2,\{3,4\}) = 10 + 25 \\ c_{13} + cost(3,\{2,4\}) = 15 + 25 \\ c_{14} + cost(4,\{2,3\}) = 20 + 24 \end{cases}$$

Mathematical Steps

$$Cost(i,S) = \min_{j\in S}\{c_{ij} + Cost(j, S-\{J\})\}$$

$Cost(2,\Phi) = 5,\quad Cost(3,\Phi) = 6,\quad Cost(4,\Phi) = 8,$

$$Cost(3,\{4\}) = c_{34} + cost(4,\Phi) = 20$$

$$Cost(4,\{3\}) = c_{43} + cost(3,\Phi) = 15$$

$$Cost(2,\{4\}) = c_{24} + cost(4,\Phi) = 18$$

$$Cost(4,\{2\}) = c_{42} + cost(2,\Phi) = 13$$

$$Cost(2,\{3\}) = c_{23} + cost(3,\Phi) = 16$$

$$Cost(3,\{2\}) = c_{32} + cost(2,\Phi) = 18$$

1 2 4 3 1

$$Cost(2,\{3{,}4\}) = \min\begin{cases} c_{23} + cost(3{,}4) = 9 + 20 \\ c_{24} + cost(4{,}3) = 10 + 15 \end{cases}$$

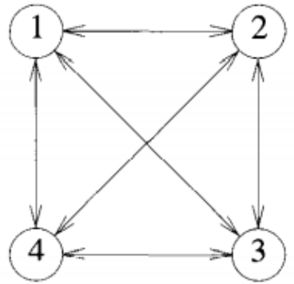
$$Cost(3,\{2{,}4\}) = \min\begin{cases} c_{32} + cost(2{,}4) = 13 + 18 \\ c_{34} + cost(4{,}2) = 12 + 13 \end{cases}$$

$$Cost(4,\{2{,}3\}) = \min\begin{cases} c_{42} + cost(2{,}3) = 8 + 16 \\ c_{43} + cost(3{,}2) = 9 + 18 \end{cases}$$

$$Cost(1,\{2{,}3{,}4\}) = \min\begin{cases} c_{12} + cost(2,\{3{,}4\}) = 10 + 25 \\ c_{13} + cost(3,\{2{,}4\}) = 15 + 25 \\ c_{14} + cost(4,\{2{,}3\}) = 20 + 24 \end{cases}$$

Question

The cost to visit from one to another city is given in the graph. Find the optimal route to visit from node 1 to all other once node and then come back to 1



$$\begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

$$g(2,\phi) = c_{21} = 5, g(3,\phi) = c_{31} = 6, \text{ and } g(4,\phi) = c_{41} = 8.$$

$$\begin{aligned} g(2,\{3\}) &= c_{23} + g(3,\phi) = 15 \\ g(3,\{2\}) &= 18 \\ g(4,\{2\}) &= 13 \end{aligned} \qquad \begin{aligned} g(2,\{4\}) &= 18 \\ g(3,\{4\}) &= 20 \\ g(4,\{3\}) &= 15 \end{aligned}$$

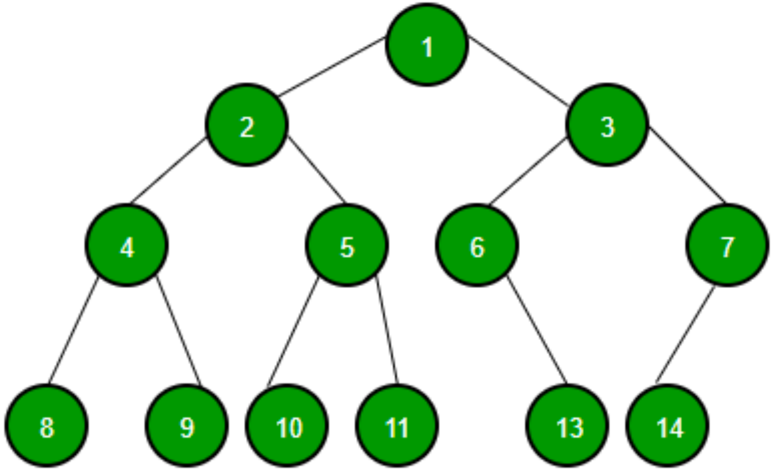
$$\begin{aligned} g(2,\{3,4\}) &= \min\ \{c_{23} + g(3,\{4\}), c_{24} + g(4,\{3\})\} = 25 \\ g(3,\{2,4\}) &= \min\ \{c_{32} + g(2,\{4\}), c_{34} + g(4,\{2\})\} = 25 \\ g(4,\{2,3\}) &= \min\ \{c_{42} + g(2,\{3\}), c_{43} + g(3,\{2\})\} = 23 \end{aligned}$$

$$\begin{aligned} g(1,\{2,3,4\}) &= \min\{c_{12} + g(2,\{3,4\}), c_{13} + g(3,\{2,4\}), c_{14} + g(4,\{2,3\})\} \\ &= \min\{35, 40, 43\} \\ &= 35 \end{aligned}$$

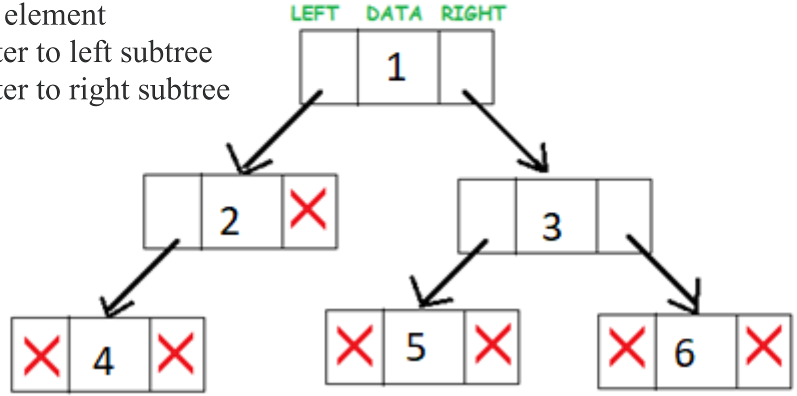
Binary Tree

Definition:

A binary tree is a hierarchical data structure in which each node has at most two children generally referred as left child and right child.



1. Data element
2. Pointer to left subtree
3. Pointer to right subtree



Traversal

When the search necessarily involves the examination of every vertex in the object being searched, it is called a traversal.

Traversal Techniques

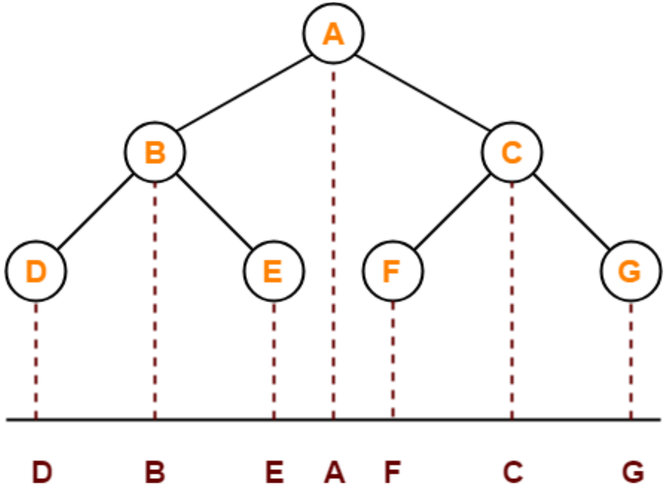
- Preorder
- Inorder
- Postorder

Inorder Trees Traversal

Approach

In this traversal method, the left subtree is visited first, then the root and later the right sub-tree.

```
Algorithm InOrder(t)
// t is a binary tree. Each node of t has
// three fields: lchild, data, and rchild.
{
    if t ≠ 0 then
    {
        InOrder(t → lchild);
        Visit(t);
        InOrder(t → rchild);
    }
}
```



Inorder Traversal : D , B , E , A , F , C , G

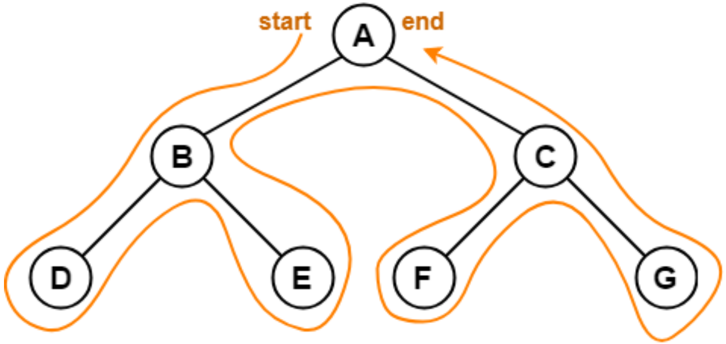
Preorder Trees Traversal

Approach

In PreOrder traversal,each node is processed before either of its sub-trees.In simpler words,Visit each node before its children.

Algorithm PreOrder(t)
// t is a binary tree. Each node of t has
// three fields: $lchild$, $data$, and $rchild$.

```
{
    if t ≠ 0 then
    {
        Visit(t);
        PreOrder(t → lchild);
        PreOrder(t → rchild);
    }
}
```



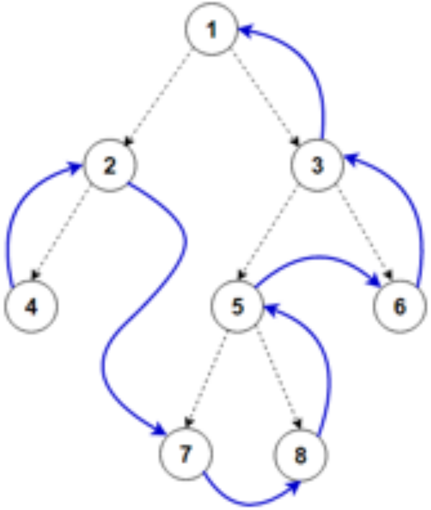
Preorder Traversal : A , B , D , E , C , F , G

Postorder Trees Traversal

Approach

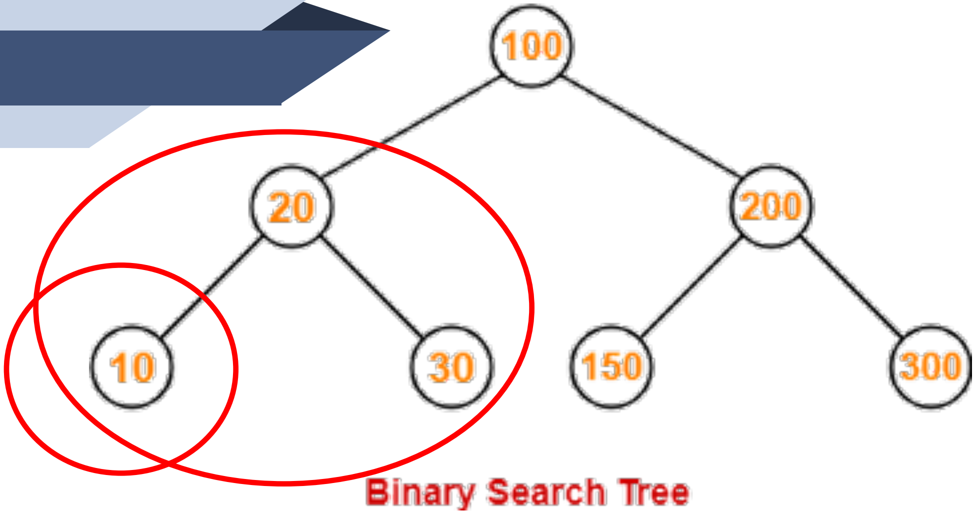
In this traversal method, the left subtree is visited first, then the root and later the right sub-tree.

Algorithm PostOrder(t)
// t is a binary tree. Each node of t has
// three fields: *lchild*, *data*, and *rchild*.
{
    if $t \neq 0$ **then**
    {
        PostOrder($t \rightarrow lchild$);
        PostOrder($t \rightarrow rchild$);
        Visit(t);
    }
}



Postorder: 4, 2, 7, 8, 5, 6, 3, 1

Example



Preorder Traversal- 100 , 20 , 10 , 30 , 200 , 150 , 300

Inorder Traversal- 10 , 20 , 30 , 100 , 150 , 200 , 300

Postorder Traversal- 10 , 30 , 20 , 150 , 300 , 200 , 100

Question

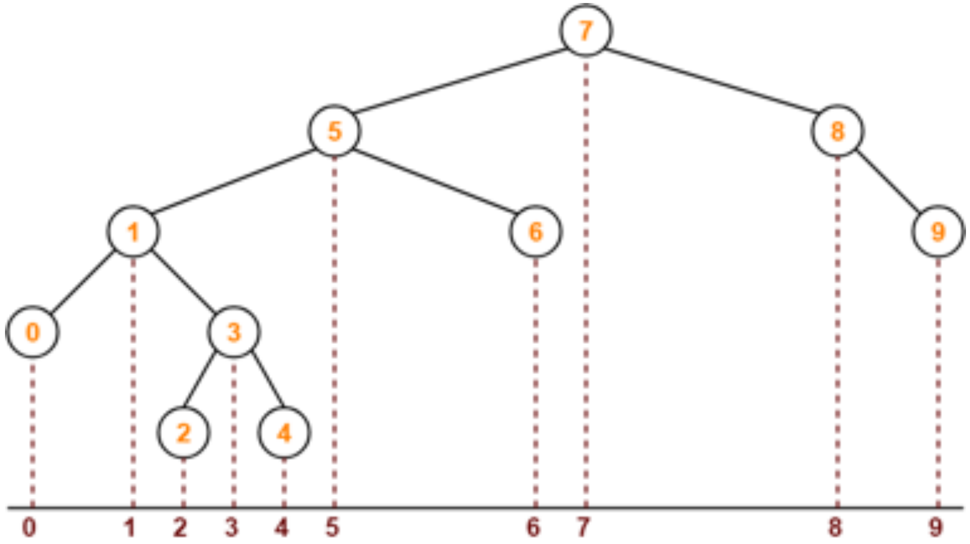
Suppose the numbers 7 , 5 , 1 , 8 , 3 , 6 , 0 , 9 , 4 , 2 are inserted in that order into an initially empty binary search tree. What is the inorder traversal sequence of the resultant tree?

A. 7 , 5 , 1 , 0 , 3 , 2 , 4 , 6 , 8 , 9

B. 0 , 2 , 4 , 3 , 1 , 6 , 5 , 9 , 8 , 7

C. 0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9

D. 9 , 8 , 6 , 4 , 2 , 3 , 0 , 1 , 5 , 7



Note: *Inorder traversal of a binary search tree always yields all the nodes in increasing order.*

Question

Construct the binary tree if the preorder and inorder sequence is known.

```
Inorder Traversal:  {        7, 5, 8, 3, 6 }
Preorder Traversal: {        3, 5, 7, 8, 6 }
```