	O(1) Iterat ©(N) O(N) Re ©(tsingdal) ion B	1219°5'82116'A''	ee (BST) Operations	: Time & Space Complexity		
Deletion	Iterative	performance				
Operation	O(N) Balanced BST	Skewed BST	Iterative vs Recursive	Best Approach & Why?	Time Complexity	Space Complexity
Insertion		•	•		-	•

	Recursion can cau	se Statos blyer Od(N) iteration is
·	Iterative	preferred

Iterative O(1) Iterat(0(eV) Φ(N) RecOn(thsigeN)

Search

O(log N) / O(N)

Iteration prevents unnecessary stack usage

		O(1) Iterative /						
	O(N) Recursive		ecursive					
Inorder Traversal		0(N) KØK(Aski) uma	tue alre	cur	sive	callsOa(Nc) maintai	ns order
Preorder Traversa	ı	0(N) O (Mi pasi	ėwreto im	ıple	mer	nt du ©(⋈) ree strud	ture
Postorder Traversa		0(N .5 809(1911): tro s	i m plem	ent	due	to s @(tk) based p	rocessing
Level Order			O(Ne)sesiv@ueue ins		tead of recursion for b		oetter	
		0(N)			effi	cien © (N)	

When to Use What?

Scenario		Best Approach		Why?
Deep BST with high depth		Avoids stack loomen floor and excessive r	ecursion calls	
Shallow BST with small size		Easier and Roccenseivaclable with fewer	unction calls	
Tree traversal		Natural aRnechsimsiphe recursive imple	mentation	
Level order traversal	Queue	based approach ilserrativeefficient and avoid	s recursion depth issues	