- Methods

|  |  |
| --- | --- |
| **No** | **Method & Description** |
| 1 | binary\_search\_tree::isEmpty()  • This method returns if binary search tree is empty. |
| 2 | binary\_search\_tree::insert(value)  • This method inserts the specified element into this binary search tree. |
| 3 | binary\_search\_tree::minValue(value)  • This method finds the node which has the minimum data among the value’s child. |
| 4 | binary\_search\_tree::maxValue(value)  • This method finds the node which has the maximum data among the value’s child. |
| 5 | binary\_search\_tree::find(value)  • This method finds the node from the tree of which data is same as value. |
| 6 | binary\_search\_tree::delete(value)  • This method is removes the node from the tree of which data is same as value. |
| 7 | binary\_search\_tree::inOrder(value)  • This method is used for tree traversal. Traversal order is left -> parent -> right. |
| 8 | binary\_search\_tree::state()  • This method shows state of this binary search tree. |
| 9 | binary\_search\_tree::clear(value)  • This method nullifies this binary search tree and make all variables initial. |

1. binary\_search\_tree::delete(value)

* Description

This delete(value) method removes the node from the tree of which data is same as value

* Declare

Following is the declaration binary\_search\_tree.pop() method.

|  |
| --- |
| var bst = new BinarySearchTree();  bst.pop(); |

* Parameter

value - The data of node which is in the binary search tree.

* Return Value

NA

* Exception

The method returns NA if node of which data is value is not among binary search tree.

* Example

|  |
| --- |
|  |

1. binary\_search\_tree::inOrder(value)

* Description

This inOrder(value) method is used for tree traversal. Traversal order is left -> parent -> right.

* Declare

Following is the declaration binary\_search\_tree.inOrder(value) method.

|  |
| --- |
| var bst = new BinarySearchTree();  bst.inOrder(this.root); |

* Parameter

value – The node is used to tree traversal recursively.

* Return Value

The method returns null if node has no data.

* Exception

The method returns null if node has no data.

* Example

|  |
| --- |
|  |

1. binary\_search\_tree::state()

* Description

This state() method shows state of the binary\_search\_tree.

* Declare

Following is the declaration binary\_search\_tree.state() method.

|  |
| --- |
| var bst = new BinarySearchTree();  bst.state(); |

* Parameter

NA

* Return Value

NA

* Exception

NA

* Example

|  |
| --- |
|  |

1. binary\_search\_tree::clear(value)

* Description

This clear() method nullifies this binary search tree.

* Declare

Following is the declaration binary\_search\_tree.clear() method.

|  |
| --- |
| var bst = new BinarySearchTree();  bst.clear(this.root); |

* Parameter

value – Recursively removes the children of the value from the tree.

* Return Value

The method return NA if node is null.

* Exception

NA

* Example

|  |
| --- |
|  |