Amortized_Weight-Balanced_Trees

1.0

Generated by Doxygen 1.8.16

1 Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

2 Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

3 Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

4 File Index

4.1 File List

Here is a list of all files with brief descriptions:

5 Namespace Documentation

5.1 BalancedBSTSet Namespace Reference

Classes

- class BalancedBSTSet

  Binary search tree implementation of the Collections interface.

Functions

- def cmp (x, y)

  Compare two objects.
- def generateRandomArray (n, vrange)

  Generates an array with a random size, filled with random elements.
- def main (args=None)

  Main function for testing.

5.1.1 Detailed Description

Amortized Weight-Balanced Trees.

Author

Paulo Roma Cavalcanti

Date

23/09/2018

5.1.2 Function Documentation

5.1.2.1 cmp()

```
def BalancedBSTSet.cmp (
        x,
        y )
```

Compare two objects.

Returns

(x > y) - (x < y)

```
Replacement for built-in function cmp that was removed in Python 3.

Compare the two objects x and y and return an integer according to
the outcome. The return value is negative if x < y, zero if x == y
and strictly positive if x > y.
```

Referenced by BalancedBSTSet.BalancedBSTSet.Node.compareTo().

5.1.2.2 generateRandomArray()

```
def BalancedBSTSet.generateRandomArray (
        n,
        vrange )
```

Generates an array with a random size, filled with random elements.

Parameters

n	maximum array size.
vrange	interval to choose the random elements from.

Returns

an array.

Referenced by main().

5.1.2.3 main()

```
def BalancedBSTSet.main (
        args = None )
```

Main function for testing.

args not used.

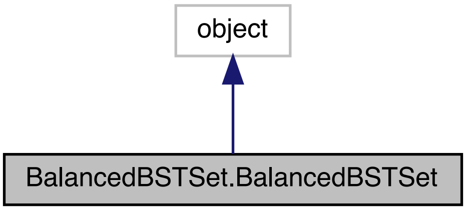
References generateRandomArray().

6 Class Documentation

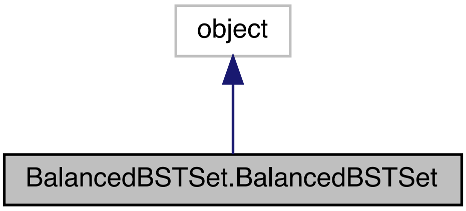
6.1 BalancedBSTSet.BalancedBSTSet Class Reference

Binary search tree implementation of the Collections interface.

Inheritance diagram for BalancedBSTSet.BalancedBSTSet:



Collaboration diagram for BalancedBSTSet.BalancedBSTSet:



Classes

- class BSTIterator

  Iterator implementation for this binary search tree.
- class Node

  Node type for this implementation.

Public Member Functions

- def __init__ (self, isSelfBalancing=False, top=0, bottom=0)

  Constructs an empty binary search tree.
- def root (self)

  Returns a read-only view of the root node of this tree.
- def isEmpty (self)

  Return whether this tree is empty.
- def rebalance (self, bstNode)

  Performs a rebalance operation on the given subtree.
- def __contains__ (self, obj)

  Returns whether the given object is in this tree.
- def add (self, key)

  Adds the given object to this tree.
- def update (self, lst)

  Adds an iterable to the tree.
- def append (self, n)

  like lists.
- def remove (self, obj)

  Removes the given object from this tree.
- def findEntry (self, key)

  Returns the node containing key, or None if the key is not found in the tree.
- def successor (self, n)

  Returns the successor of the given node.
- def antecessor (self, n)

  Returns the antecessor of the given node.
- def unlinkNode (self, n)

Removes the given node, preserving the binary search tree property of the tree.

- def iterator (self)

  Returns an iterator for this tree.
- def __len__ (self)

  Returns the number of elements in this tree.
- def toArray (self)

  Returns an array containing all of the elements in this tree.
- def __getitem__ (self, ind)

  Indexing operator [].
- def __iter__ (self)

  Iterator as a generator.
- def __reversed__ (self)

  Reversed Iterator.
- def height (self)

  Return the height of this tree.
- def getHeight (self, root)

  Return the height of a subtree.
- def __repr__ (self)

  Returns a representation of this tree as a multi-line string.
- def __str__ (self)

  Prints the nodes of this tree in order.

Private Member Functions

- def __rebalanceRec (self, arr, start, end)

  Recursively creates a tree from a sorted array of nodes.
- def __inOrder (self, node, arr)

  Executes an in order traversal of the tree rooted at a given node.
- def __incrementNodesUp (self, n)

  Increments all node counters in the path to the root.
- def __decrementNodesUp (self, n)

  Decrements all node counters in the path to the root.
- def __toStringRec (self, n, sb, depth)

  Preorder traversal of the tree that builds a string representation in the given StringBuilder.

Private Attributes

- __root

  Root of this tree.
- __size

  Number of elements in this tree.
- __doBalance

  Turns balancing on or off.
- __numerator

  Numerator of the self-balancing tree's α value.
- __denominator

  Denominator of the self-balancing tree's α value.

6.1.1 Detailed Description

Binary search tree implementation of the Collections interface.

- The __contains__() and remove() methods of Collections Abstract Base Classes are overridden to search the tree without using the iterator.
- Instances of this class always maintain node counts that is, the Node.counter variable of the Node interface is calculated in O(1).
- If constructed with the isSelfBalancing flag True, instances of this tree are self-balancing:
  - whenever an add(), remove(), or BSTIterator.remove() operation causes any node to become unbalanced,
    a rebalance operation is automatically performed at the highest unbalanced node.

  To run:

  - python BalancedBSTSet.py

Author

Paulo Roma Cavalcanti

Since

23/09/2018

See also

`Scapegoat tree`
`Original paper`
`Collections Abstract Base Classes`
`Amortized weight-balanced trees`

6.1.2 Constructor & Destructor Documentation

6.1.2.1 __init__()

```
def BalancedBSTSet.BalancedBSTSet.__init__ (
        self,
        isSelfBalancing = False,
        top = 0,
        bottom = 0 )
```

Constructs an empty binary search tree.

If the isSelfBalancing flag is True, the tree will be self-balancing: if so, whenever an add(), remove(), or BSTIterator.remove() operation causes any node to become unbalanced, a rebalance operation is automatically performed at the highest unbalanced node.

The given $\frac{1}{2} \leq \alpha = \frac{top}{bottom} < 1$ is used for the balance condition. An α of 1, therefore, would describe a linked list as balanced, whereas an α of 0.5 would only match almost complete binary trees.

Maintains node counts whether or not isSelfBalancing is True.

Parameters

isSelfBalancing	True if this binary search tree is to be self-balancing, and False otherwise.
top	numerator of the fraction α.
bottom	denominator of the fraction α.

Exceptions

AttributeError	if $\frac{top}{bottom} < \frac{1}{2}$

6.1.3 Member Function Documentation

6.1.3.1 __contains__()

```
def BalancedBSTSet.BalancedBSTSet.__contains__ (
            self,
            obj )
```

Returns whether the given object is in this tree.

Parameters

obj	given object.

Returns

True if the object is in the tree, or False otherwise.

References BalancedBSTSet.BalancedBSTSet.findEntry().

6.1.3.2 __decrementNodesUp()

```
def BalancedBSTSet.BalancedBSTSet.__decrementNodesUp (
            self,
            n ) [private]
```

Decrements all node counters in the path to the root.

Parameters

n	reference node.

Returns

first (higher) unbalanced node, or None if all nodes are balanced.

References BalancedBSTSet.BalancedBSTSet.__denominator, and BalancedBSTSet.BalancedBSTSet.__↵ numerator.

Referenced by BalancedBSTSet.BalancedBSTSet.unlinkNode().

6.1.3.3 __getitem__()

```
def BalancedBSTSet.BalancedBSTSet.__getitem__ (
            self,
            ind )
```

Indexing operator [].

Exceptions

IndexError.	

Parameters

ind	index to retrieve.

Returns

ind-ith value in the tree, or an exception.

References BalancedBSTSet.BalancedBSTSet.__size, and BalancedBSTSet.BalancedBSTSet.iterator().

6.1.3.4 __incrementNodesUp()

```
def BalancedBSTSet.BalancedBSTSet.__incrementNodesUp (
            self,
            n )  [private]
```

Increments all node counters in the path to the root.

Parameters

n	reference node.

Returns

first (higher) unbalanced node, or None if all nodes are balanced.

References BalancedBSTSet.BalancedBSTSet.__denominator, and BalancedBSTSet.BalancedBSTSet.__↵ numerator.

Referenced by BalancedBSTSet.BalancedBSTSet.add().

6.1.3.5 __inOrder()

```
def BalancedBSTSet.BalancedBSTSet.__inOrder (
        self,
        node,
        arr ) [private]
```

Executes an in order traversal of the tree rooted at a given node.

Parameters

node	root.
arr	array for holding the node data.

Returns

arr.

References BalancedBSTSet.BalancedBSTSet.__inOrder().

Referenced by BalancedBSTSet.BalancedBSTSet.__inOrder(), and BalancedBSTSet.BalancedBSTSet.↩
rebalance().

6.1.3.6 __iter__()

```
def BalancedBSTSet.BalancedBSTSet.__iter__ (
        self )
```

Iterator as a generator.

Generators are functions having an yield keyword. Any function which has “yield” in it is a generator.

Generator takes care of creating the iterable. It also takes care of creating the underlying iterator. And next() of this iterator() is such that it returns each ‘yield’

See also

https://www.agiliq.com/blog/2017/11/how-python-generators-are-similar-iterators/

References BalancedBSTSet.BalancedBSTSet.iterator().

6.1.3.7 __len__()

```
def BalancedBSTSet.BalancedBSTSet.__len__ (
        self )
```

Returns the number of elements in this tree.

References BalancedBSTSet.BalancedBSTSet.__size.

6.1.3.8 __rebalanceRec()

```
def BalancedBSTSet.BalancedBSTSet.__rebalanceRec (
        self,
        arr,
        start,
        end ) [private]
```

Recursively creates a tree from a sorted array of nodes.

Parameters

arr	array of nodes.
start	beginning of the array.
end	end of the array.

Returns

root of the tree.

See also

https://articles.leetcode.com/convert-sorted-array-into-balanced/

References BalancedBSTSet.BalancedBSTSet.__rebalanceRec().

Referenced by BalancedBSTSet.BalancedBSTSet.__rebalanceRec(), and BalancedBSTSet.BalancedBSTSet.↩ rebalance().

6.1.3.9 __repr__()

```
def BalancedBSTSet.BalancedBSTSet.__repr__ (
        self )
```

Returns a representation of this tree as a multi-line string.

The tree is drawn with the root at the left and children are shown top-to-bottom. Leaves are marked with a "-" and non-leaves are marked with a "+".

References BalancedBSTSet.BalancedBSTSet.__root, and BalancedBSTSet.BalancedBSTSet.__toStringRec().

6.1.3.10 __reversed__()

```
def BalancedBSTSet.BalancedBSTSet.__reversed__ (
        self )
```

Reversed Iterator.

References BalancedBSTSet.BalancedBSTSet.iterator().

6.1.3.11 __str__()

```
def BalancedBSTSet.BalancedBSTSet.__str__ (
        self )
```

Prints the nodes of this tree in order.

6.1.3.12 __toStringRec()

```
def BalancedBSTSet.BalancedBSTSet.__toStringRec (
        self,
        n,
        sb,
        depth ) [private]
```

Preorder traversal of the tree that builds a string representation in the given StringBuilder.

Parameters

n	root of subtree to be traversed.
sb	list in which to create a string representation.
depth	depth of the given node in the tree.

References BalancedBSTSet.BalancedBSTSet.__toStringRec().

Referenced by BalancedBSTSet.BalancedBSTSet.__repr__(), and BalancedBSTSet.BalancedBSTSet.__to↩ StringRec().

6.1.3.13 add() `def BalancedBSTSet.BalancedBSTSet.add ( self, key )`

Adds the given object to this tree.

Parameters

key	given object.

Returns

True if the object was found, and False otherwise.

References BalancedBSTSet.BalancedBSTSet.__doBalance, BalancedBSTSet.BalancedBSTSet.__increment↩ NodesUp(), BalancedBSTSet.BalancedBSTSet.__root, BalancedBSTSet.BalancedBSTSet.__size, and Balanced↩ BSTSet.BalancedBSTSet.rebalance().

Referenced by BalancedBSTSet.BalancedBSTSet.append(), and BalancedBSTSet.BalancedBSTSet.update().

6.1.3.14 antecessor() `def BalancedBSTSet.BalancedBSTSet.antecessor ( self, n )`

Returns the antecessor of the given node.

Parameters

n	a node.

Returns

the antecessor of the given node in this tree, or None if there is no antecessor.

6.1.3.15 append() `def BalancedBSTSet.BalancedBSTSet.append (`

```
self,
n )
```

like lists.

References BalancedBSTSet.BalancedBSTSet.add().

6.1.3.16 findEntry() `def BalancedBSTSet.BalancedBSTSet.findEntry (`

```
self,
key )
```

Returns the node containing key, or None if the key is not found in the tree.

Parameters

key	

Returns

the node containing key, or None if not found.

References BalancedBSTSet.BalancedBSTSet.__root.

Referenced by BalancedBSTSet.BalancedBSTSet.__contains__(), and BalancedBSTSet.BalancedBSTSet.remove().

6.1.3.17 getHeight() `def BalancedBSTSet.BalancedBSTSet.getHeight (`

```
self,
root )
```

Return the height of a subtree.

The height of a node is the number of edges on the longest path between that node and a leaf. The height of a leaf is 0.

Parameters

root	node of the subtree.

References BalancedBSTSet.BalancedBSTSet.getHeight().

Referenced by BalancedBSTSet.BalancedBSTSet.getHeight(), and BalancedBSTSet.BalancedBSTSet.height().

6.1.3.18 height() `def BalancedBSTSet.BalancedBSTSet.height (`

```
self )
```

Return the height of this tree.

The height of a tree is the height of its root node.

References BalancedBSTSet.BalancedBSTSet.__root, and BalancedBSTSet.BalancedBSTSet.getHeight().

6.1.3.19 isEmpty() `def BalancedBSTSet.BalancedBSTSet.isEmpty (`

```
self )
```

Return whether this tree is empty.

References BalancedBSTSet.BalancedBSTSet.__root.

6.1.3.20 iterator() `def BalancedBSTSet.BalancedBSTSet.iterator (`

```
self )
```

Returns an iterator for this tree.

Referenced by BalancedBSTSet.BalancedBSTSet.__getitem__(), BalancedBSTSet.BalancedBSTSet.__iter__↩ (), BalancedBSTSet.BalancedBSTSet.__reversed__(), and BalancedBSTSet.BalancedBSTSet.toArray().

6.1.3.21 rebalance() `def BalancedBSTSet.BalancedBSTSet.rebalance (`

```
self,
bstNode )
```

Performs a rebalance operation on the given subtree.

This operation does not create or destroy any nodes and does not change the size of this tree.

Parameters

bstNode	root of the subtree to be rebalanced.

References BalancedBSTSet.BalancedBSTSet.__inOrder(), BalancedBSTSet.BalancedBSTSet.__rebalance↩ Rec(), and BalancedBSTSet.BalancedBSTSet.__root.

Referenced by BalancedBSTSet.BalancedBSTSet.add(), and BalancedBSTSet.BalancedBSTSet.unlinkNode().

6.1.3.22 remove() `def BalancedBSTSet.BalancedBSTSet.remove (`

```
self,
obj )
```

Removes the given object from this tree.

Parameters

obj	given object.

Returns

True if the object was found, and False otherwise.

References BalancedBSTSet.BalancedBSTSet.findEntry(), and BalancedBSTSet.BalancedBSTSet.unlinkNode().

6.1.3.23 root()

```
def BalancedBSTSet.BalancedBSTSet.root (
        self )
```

Returns a read-only view of the root node of this tree.

Returns

root node of this tree.

References BalancedBSTSet.BalancedBSTSet.__root.

6.1.3.24 successor()

```
def BalancedBSTSet.BalancedBSTSet.successor (
        self,
        n )
```

Returns the successor of the given node.

Parameters

n	

Returns

the successor of the given node in this tree, or None if there is no successor.

Referenced by BalancedBSTSet.BalancedBSTSet.unlinkNode().

6.1.3.25 toArray()

```
def BalancedBSTSet.BalancedBSTSet.toArray (
        self )
```

Returns an array containing all of the elements in this tree.

If the collection makes any guarantees as to what order its elements are returned by its iterator, this method must return the elements in the same order.

Returns

a list of node data (keys).

References BalancedBSTSet.BalancedBSTSet.iterator().

6.1.3.26 unlinkNode() `def BalancedBSTSet.BalancedBSTSet.unlinkNode (`

```
self,
n )
```

Removes the given node, preserving the binary search tree property of the tree.

Parameters

n	node to be removed.

References BalancedBSTSet.BalancedBSTSet.__decrementNodesUp(), BalancedBSTSet.BalancedBSTSet.__doBalance, BalancedBSTSet.BalancedBSTSet.__root, BalancedBSTSet.BalancedBSTSet.__size, BalancedBSTSet.BalancedBSTSet.rebalance(), and BalancedBSTSet.BalancedBSTSet.successor().

Referenced by BalancedBSTSet.BalancedBSTSet.remove(), and BalancedBSTSet.BalancedBSTSet.BSTIterator.remove().

6.1.3.27 update() `def BalancedBSTSet.BalancedBSTSet.update (`

```
self,
lst )
```

Adds an iterable to the tree.

References BalancedBSTSet.BalancedBSTSet.add().

6.1.4 Member Data Documentation

6.1.4.1 __denominator `BalancedBSTSet.BalancedBSTSet.__denominator [private]`

Denominator of the self-balancing tree's α value.

Referenced by BalancedBSTSet.BalancedBSTSet.__decrementNodesUp(), and BalancedBSTSet.BalancedBSTSet.__incrementNodesUp().

6.1.4.2 __doBalance `BalancedBSTSet.BalancedBSTSet.__doBalance [private]`

Turns balancing on or off.

Referenced by BalancedBSTSet.BalancedBSTSet.add(), and BalancedBSTSet.BalancedBSTSet.unlinkNode().

6.1.4.3 __numerator `BalancedBSTSet.BalancedBSTSet.__numerator [private]`

Numerator of the self-balancing tree's α value.

Referenced by BalancedBSTSet.BalancedBSTSet.__decrementNodesUp(), and BalancedBSTSet.BalancedBSTSet.__incrementNodesUp().

6.1.4.4 __root `BalancedBSTSet.BalancedBSTSet.__root [private]`

Root of this tree.

Referenced by BalancedBSTSet.BalancedBSTSet.__repr__(), BalancedBSTSet.BalancedBSTSet.add(), BalancedBSTSet.BalancedBSTSet.findEntry(), BalancedBSTSet.BalancedBSTSet.height(), BalancedBSTSet.BalancedBSTSet.isEmpty(), BalancedBSTSet.BalancedBSTSet.rebalance(), BalancedBSTSet.BalancedBSTSet.root(), and BalancedBSTSet.BalancedBSTSet.unlinkNode().

6.1.4.5 __size `BalancedBSTSet.BalancedBSTSet.__size [private]`

Number of elements in this tree.

Referenced by BalancedBSTSet.BalancedBSTSet.__getitem__(), BalancedBSTSet.BalancedBSTSet.__len__(), BalancedBSTSet.BalancedBSTSet.add(), and BalancedBSTSet.BalancedBSTSet.unlinkNode().

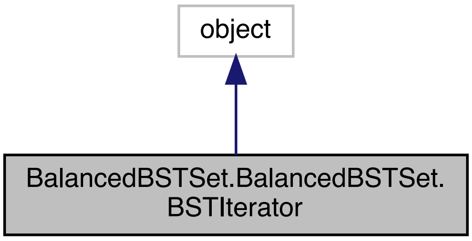
The documentation for this class was generated from the following file:

- BalancedBSTSet.py

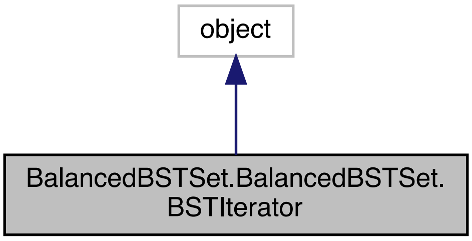
6.2 BalancedBSTSet.BalancedBSTSet.BSTIterator Class Reference

Iterator implementation for this binary search tree.

Inheritance diagram for BalancedBSTSet.BalancedBSTSet.BSTIterator:



Collaboration diagram for BalancedBSTSet.BalancedBSTSet.BSTIterator:



Public Member Functions

- def getSmallestValue (self, n)

  return the smallest value of the tree.
- def getLargestValue (self, n)

  return the largest value of the tree.
- def direction (self)

  getter for the iterator direction.
- def direction (self, dir)

  setter for this iterator direction: forward or backward.
- def __init__ (self, tree)

  Constructs an iterator starting at the smallest ot largest element in the tree.
- def __iter__ (self)

  Forward iterator.

- def __reversed__ (self)

  Reverse iterator.
- def hasNext (self)

  Whether current is not None.
- def peek (self)

  Return the content of the current node without advancing.
- def __next__ (self)

  Returns current node, which is saved in pending.
- def next (self)

  For python 2.
- def remove (self)

  Removes the node returned by the last call to next().

Public Attributes

- direction

  direction of iteration: forward or backward.

Static Public Attributes

- int forward = 0

  iterates forward.
- int backward = 1

  iterates backward.

Private Attributes

- __dir

  holds the current traversal direction.
- __current

  Node to be returned by next call to next().
- __pending

  Node returned by last call to next() and available for removal.
- __tree

  The tree to be traversed.
- __move

  move method.

6.2.1 Detailed Description

Iterator implementation for this binary search tree.

The elements are returned in ascending order according to their natural ordering.

A reversed iterator calls __reversed__() first and then __iter__().

6.2.2 Constructor & Destructor Documentation

6.2.2.1 __init__() `def BalancedBSTSet.BalancedBSTSet.BSTIterator.__init__ (`

```
self,
tree )
```

Constructs an iterator starting at the smallest ot largest element in the tree.

6.2.3 Member Function Documentation

6.2.3.1 __iter__() `def BalancedBSTSet.BalancedBSTSet.BSTIterator.__iter__ (`

```
self )
```

Forward iterator.

References BalancedBSTSet.BalancedBSTSet.BSTIterator.__move, and BalancedBSTSet.BalancedBSTSet.BSTIterator.direction.

6.2.3.2 __next__() `def BalancedBSTSet.BalancedBSTSet.BSTIterator.__next__ (`

```
self )
```

Returns current node, which is saved in pending.

Current is set to successor(current).

References BalancedBSTSet.BalancedBSTSet.BSTIterator.__current, BalancedBSTSet.BalancedBSTSet.BSTIterator.__move, BalancedBSTSet.BalancedBSTSet.BSTIterator.__pending, and BalancedBSTSet.BalancedBSTSet.BSTIterator.hasNext().

Referenced by BalancedBSTSet.BalancedBSTSet.BSTIterator.next().

6.2.3.3 __reversed__() `def BalancedBSTSet.BalancedBSTSet.BSTIterator.__reversed__ (`

```
self )
```

Reverse iterator.

References BalancedBSTSet.BalancedBSTSet.BSTIterator.direction.

6.2.3.4 direction() [1/2] `def BalancedBSTSet.BalancedBSTSet.BSTIterator.direction ( self )`

getter for the iterator direction.

References BalancedBSTSet.BalancedBSTSet.BSTIterator.direction.

6.2.3.5 direction() [2/2] `def BalancedBSTSet.BalancedBSTSet.BSTIterator.direction ( self, dir )`

setter for this iterator direction: forward or backward.

References BalancedBSTSet.BalancedBSTSet.BSTIterator.__dir, and BalancedBSTSet.BalancedBSTSet.BSTIterator.direction.

6.2.3.6 getLargestValue() `def BalancedBSTSet.BalancedBSTSet.BSTIterator.getLargestValue ( self, n )`

return the largest value of the tree.

6.2.3.7 getSmallestValue() `def BalancedBSTSet.BalancedBSTSet.BSTIterator.getSmallestValue ( self, n )`

return the smallest value of the tree.

6.2.3.8 hasNext() `def BalancedBSTSet.BalancedBSTSet.BSTIterator.hasNext ( self )`

Whether current is not None.

References BalancedBSTSet.BalancedBSTSet.BSTIterator.__current.

Referenced by BalancedBSTSet.BalancedBSTSet.BSTIterator.__next__().

6.2.3.9 next() `def BalancedBSTSet.BalancedBSTSet.BSTIterator.next ( self )`

For python 2.

References BalancedBSTSet.BalancedBSTSet.BSTIterator.__next__().

6.2.3.10 peek() `def BalancedBSTSet.BalancedBSTSet.BSTIterator.peek ( self )`

Return the content of the current node without advancing.

References BalancedBSTSet.BalancedBSTSet.BSTIterator.__current.

6.2.3.11 remove() `def BalancedBSTSet.BalancedBSTSet.BSTIterator.remove ( self )`

Removes the node returned by the last call to next().

Current pos to the successor of pending, but if pending has two children, then unlinkNode(pending) will copy the successor's data of pending and delete the successor node. So in this case, we want to end up with current pointing to the pending node.

References BalancedBSTSet.BalancedBSTSet.BSTIterator.__current, BalancedBSTSet.BalancedBSTSet.BST↩ Iterator.__pending, BalancedBSTSet.BalancedBSTSet.BSTIterator.__tree, and BalancedBSTSet.BalancedBST↩ Set.unlinkNode().

6.2.4 Member Data Documentation

6.2.4.1 __current `BalancedBSTSet.BalancedBSTSet.BSTIterator.__current [private]`

Node to be returned by next call to next().

Referenced by BalancedBSTSet.BalancedBSTSet.BSTIterator.__next__(), BalancedBSTSet.BalancedBSTSet.B↩ STIterator.hasNext(), BalancedBSTSet.BalancedBSTSet.BSTIterator.peek(), and BalancedBSTSet.BalancedBS↩ TSet.BSTIterator.remove().

6.2.4.2 __dir `BalancedBSTSet.BalancedBSTSet.BSTIterator.__dir [private]`

holds the current traversal direction.

Referenced by BalancedBSTSet.BalancedBSTSet.BSTIterator.direction().

6.2.4.3 __move `BalancedBSTSet.BalancedBSTSet.BSTIterator.__move [private]`

move method.

Referenced by BalancedBSTSet.BalancedBSTSet.BSTIterator.__iter__(), and BalancedBSTSet.BalancedBST↩ Set.BSTIterator.__next__().

6.2.4.4 __pending `BalancedBSTSet.BalancedBSTSet.BSTIterator.__pending [private]`

Node returned by last call to next() and available for removal.

This field is None when no node is available to be removed.

Referenced by BalancedBSTSet.BalancedBSTSet.BSTIterator.__next__(), and BalancedBSTSet.BalancedBST↩Set.BSTIterator.remove().

6.2.4.5 __tree `BalancedBSTSet.BalancedBSTSet.BSTIterator.__tree [private]`

The tree to be traversed.

Inner classes do not have access to outer class variables.

Referenced by BalancedBSTSet.BalancedBSTSet.BSTIterator.remove().

6.2.4.6 backward `int BalancedBSTSet.BalancedBSTSet.BSTIterator.backward = 1 [static]`

iterates backward.

6.2.4.7 direction `BalancedBSTSet.BalancedBSTSet.BSTIterator.direction`

direction of iteration: forward or backward.

Referenced by BalancedBSTSet.BalancedBSTSet.BSTIterator.__iter__(), BalancedBSTSet.BalancedBSTSet.B↩STIterator.__reversed__(), and BalancedBSTSet.BalancedBSTSet.BSTIterator.direction().

6.2.4.8 forward `int BalancedBSTSet.BalancedBSTSet.BSTIterator.forward = 0 [static]`

iterates forward.

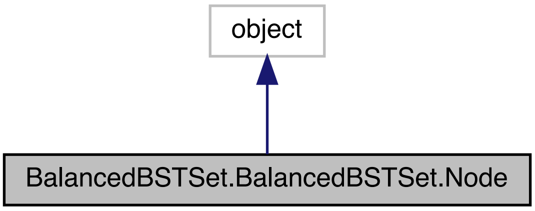
The documentation for this class was generated from the following file:

- BalancedBSTSet.py

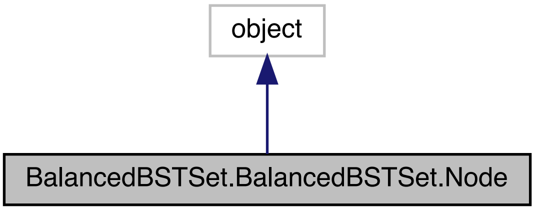
6.3 BalancedBSTSet.BalancedBSTSet.Node Class Reference

Node type for this implementation.

Inheritance diagram for BalancedBSTSet.BalancedBSTSet.Node:



Collaboration diagram for BalancedBSTSet.BalancedBSTSet.Node:



Public Member Functions

- def __init__ (self, key, parent)

  Constructor given a data object and the parent of this node.
- def __str__ (self)

  Return a string representation of this node.
- def __repr__ (self)

  Return a string representation of this node.
- def isUnBalanced (self, num, den)

  Returns whether this node is not balanced.
- def compareTo (self, key)

  Compares the data of this node to a given key.

Public Attributes

- data

  Data (object) in this node.
- parent

  Reference to the parent node.
- counter

  Number of nodes in the subtree rooted at this node.
- left

  Reference to the left child node.
- right

  Reference to the right child node.

6.3.1 Detailed Description

Node type for this implementation.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 __init__()

```
def BalancedBSTSet.BalancedBSTSet.Node.__init__ (
        self,
        key,
        parent )
```

Constructor given a data object and the parent of this node.

Parameters

key	data object.
parent	parent node.

6.3.3 Member Function Documentation

6.3.3.1 __repr__()

```
def BalancedBSTSet.BalancedBSTSet.Node.__repr__ (
        self )
```

Return a string representation of this node.

References BalancedBSTSet.BalancedBSTSet.Node.counter, and BalancedBSTSet.BalancedBSTSet.Node.data.

6.3.3.2 __str__() `def BalancedBSTSet.BalancedBSTSet.Node.__str__ ( self )`

Return a string representation of this node.

References BalancedBSTSet.BalancedBSTSet.Node.data.

6.3.3.3 compareTo() `def BalancedBSTSet.BalancedBSTSet.Node.compareTo ( self, key )`

Compares the data of this node to a given key.

Returns

1 if the data of this object is greater than key's,
-1 if the data of this object is smaller than key's or
0 if it is equal

References BalancedBSTSet.cmp(), and BalancedBSTSet.BalancedBSTSet.Node.data.

6.3.3.4 isUnBalanced() `def BalancedBSTSet.BalancedBSTSet.Node.isUnBalanced ( self, num, den )`

Returns whether this node is not balanced.

References BalancedBSTSet.BalancedBSTSet.Node.counter, BalancedBSTSet.BalancedBSTSet.Node.left, and BalancedBSTSet.BalancedBSTSet.Node.right.

6.3.4 Member Data Documentation

6.3.4.1 counter `BalancedBSTSet.BalancedBSTSet.Node.counter`

Number of nodes in the subtree rooted at this node.

Referenced by BalancedBSTSet.BalancedBSTSet.Node.__repr__(), and BalancedBSTSet.BalancedBSTSet.↩ Node.isUnBalanced().

6.3.4.2 data `BalancedBSTSet.BalancedBSTSet.Node.data`

Data (object) in this node.

Referenced by BalancedBSTSet.BalancedBSTSet.Node.__repr__(), BalancedBSTSet.BalancedBSTSet.Node.__str__(), and BalancedBSTSet.BalancedBSTSet.Node.compareTo().

6.3.4.3 left `BalancedBSTSet.BalancedBSTSet.Node.left`

Reference to the left child node.

Referenced by BalancedBSTSet.BalancedBSTSet.Node.isUnBalanced().

6.3.4.4 parent `BalancedBSTSet.BalancedBSTSet.Node.parent`

Reference to the parent node.

6.3.4.5 right `BalancedBSTSet.BalancedBSTSet.Node.right`

Reference to the right child node.

Referenced by BalancedBSTSet.BalancedBSTSet.Node.isUnBalanced().

The documentation for this class was generated from the following file:

- BalancedBSTSet.py

7 File Documentation

7.1 BalancedBSTSet.py File Reference

Classes

- class BalancedBSTSet.BalancedBSTSet

  Binary search tree implementation of the Collections interface.
- class BalancedBSTSet.BalancedBSTSet.Node

  Node type for this implementation.
- class BalancedBSTSet.BalancedBSTSet.BSTIterator

  Iterator implementation for this binary search tree.

Namespaces

- BalancedBSTSet

Functions

- def BalancedBSTSet.cmp (x, y)

  Compare two objects.
- def BalancedBSTSet.generateRandomArray (n, vrange)

  Generates an array with a random size, filled with random elements.
- def BalancedBSTSet.main (args=None)

  Main function for testing.

Index