PROGRAMMERS: MC NGUYEN, HOANG LE, PHUC DO

EFFICIENCY

Constructor or Method Name	Efficiency	Big O Notation
SortedMapNode(K key, V value, SortedMapNode<K, V> lessThanNode, SortedMapNode<K, V> greaterThanNode)	Constant	O(1)
getKey()	Constant	O(1)
setKey(K key)	Constant	O(1)
getValue()	Constant	O(1)
setValue(V value)	Constant	O(1)
getLessThanNode()	Constant	O(1)
setLessThanNode(SortedMapNode<K, V> lessThanNode)	Constant	O(1)
getGreaterThanNode()	Constant	O(1)
setGreaterThanNode(SortedMapNode<K, V> greaterThanNode)	Constant	O(1)
SortedMap()	Constant	O(1)
SortedMap(SortedMap<K, V> sortedMap)	Best case: Constant Worst case: Linear Average case: Linear	Best case: O(1) Worst case: O(n) Average case: O(n)
getSize()	Constant	O(1)
isEmpty()	Constant	O(1)
getSmallestValue()	Constant	O(1)
getLargestValue()	Constant	O(1)
contains(K key)	Best case: Constant Worst case: Linear Average case: Logarithmic	Best case: O(1) Worst case: O(n) Average case: O(log n)

containsAll(K[] keyArray)	Best case: Constant Worst case: Linear Average case: Logarithmic	Best case: O(1) Worst case: O(n) Average case: O(log n)
add(K key, V value)	Best case: Constant Worst case: Linear Average case: Logarithmic	Best case: O(1) Worst case: O(n) Average case: O(log n)
addAll(SortedMap<K, V> sortedMap)	Best case: Constant Worst case: Quadratic Average case: Logarithmic Linear	Best case: O(1) Worst case: O(n^2) Average case: O(n log n)
removeSmallestValue()	Constant	O(1)
removeLargestValue()	Constant	O(1)
remove(K key)	Best case: Constant Worst case: Linear Average case: Logarithmic	Best case: O(1) Worst case: O(n) Average case: O(log n)
removeAll(SortedMap<K, V> sortedMap)	Best case: Constant Worst case: Quadratic Average case: Logarithmic Linear	Best case: O(1) Worst case: O(n^2) Average case: O(n log n)
replace(K key, V newValue)	Best case: Constant Worst case: Linear Average case: Logarithmic	Best case: O(1) Worst case: O(n) Average case: O(log n)
clear()	Constant	O(1)
saveToFile(String filename)	Linear	O(n)