Java - Project 2/src/ADTSortingTimer.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access | Java | Debug

Package ... | JUnit

CorrelatorResults.j
DataCount.java
DataCounter.java
DataCountStringC
FileWordReader.jav
FourHeap.java
GArrayStack.java
GStack.java
Hasher.java
Hashtable.java
MoveToFrontList.j
PriorityQueue.java
SimpleIterator.java
StringComparator.
StringHasher.java
StringHasher2.java
StringHasher3.java
TestAVLTree.java
TestBinarySearchT
TestFourHeap.java
TestGArrayStack.ja
TestHashtable.java
TestMoveToFrontL
TestSorting.java
TestStringCompar
WordCount.java
WordCount2.java
JRE System Library [JavaSE-1.7]
JUnit 4
Test

WordCount.java | ADTSortingTimer.java | StringHasher2.java | StringHasher.java | StringHasher3.java | WordCount2.java

```
/**
 * Print out the time of processing the text and printing out the word and its count in descending order
 * used by different combinations of an abstract data type and a sorting algorithm.
 *
 * @author Chun-Wei Chen
 * @version CSE 332 02/19/13
 */
public class ADTSortingTimer {

    /**
     * @param args
     */
    public static void main(String[] args) {
        if (args.length != 1) {
            System.err.println("Invalid number of arguments");
            System.exit(1);
        } else {
            long startTime, endTime;
```

Chun-Wei Chen

CSE 332

Project 2 - Write-Up Questions No.9

ADT Sorting Timing

Input file: The Advancement of Learning

Javadoc | Declaration | Console | Debug

<terminated> ADTSortingTimer [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Feb 19, 2013 6:12:11 AM)

```
Timing Results:
Time of BinarySearchTree + insertionSort: 2542 ms
Time of BinarySearchTree + heapSort: 635 ms
Time of BinarySearchTree + mergeSort: 361 ms
Time of AVLTree + insertionSort: 88245 ms
Time of AVLTree + heapSort: 71072 ms
Time of AVLTree + mergeSort: 72634 ms
Time of MoveToFrontList + insertionSort: 6833 ms
Time of MoveToFrontList + heapSort: 6632 ms
Time of MoveToFrontList + mergeSort: 4927 ms
Time of Hashtable + insertionSort: 5202 ms
Time of Hashtable + heapSort: 4800 ms
Time of Hashtable + mergeSort: 3986 ms
```

Java - Project 2/src/ADTSortingTimer.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access | Java | Debug

Package ... | JUnit

- CorrelatorResults.j
- DataCount.java
- DataCounter.java
- DataCountStringC
- FileWordReader.jav
- FourHeap.java
- GArrayStack.java
- GStack.java
- Hasher.java
- Hashtable.java
- MoveToFrontList.j
- PriorityQueue.java
- SimpleIterator.java
- StringComparator.
- StringHasher.java
- StringHasher2.java
- StringHasher3.java
- TestAVLTree.java
- TestBinarySearchT
- TestFourHeap.java
- TestGArrayStack.ja
- TestHashtable.java
- TestMoveToFrontL
- TestSorting.java
- TestStringCompar
- WordCount.java
- WordCount2.java
- JRE System Library [JavaSE
- JUnit 4
- Test

WordCount.java | ADTSortingTimer.java | StringHasher2.java | StringHasher.java | StringHasher3.java | WordCount2.java

```
/**
 * Print out the time of processing the text and printing out the word and its count in descending order
 * used by different combinations of an abstract data type and a sorting algorithm.
 *
 * @author Chun-Wei Chen
 * @version CSE 332 02/19/13
 */
public class ADTSortingTimer {

    /**
     * @param args
     */
    public static void main(String[] args) {
        if (args.length != 1) {
            System.err.println("Invalid number of arguments");
            System.exit(1);
        } else {
            long startTime, endTime;
```

Javadoc | Declaration | Console | Debug

<terminated> ADTSortingTimer [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Feb 19, 2013 6:19:52 AM)

```
Timing Results:
Time of BinarySearchTree + insertionSort: 2597 ms
Time of BinarySearchTree + heapSort: 628 ms
Time of BinarySearchTree + mergeSort: 412 ms
Time of AVLTree + insertionSort: 93224 ms
Time of AVLTree + heapSort: 68957 ms
Time of AVLTree + mergeSort: 70314 ms
Time of MoveToFrontList + insertionSort: 7274 ms
Time of MoveToFrontList + heapSort: 6586 ms
Time of MoveToFrontList + mergeSort: 5349 ms
Time of Hashtable + insertionSort: 5068 ms
Time of Hashtable + heapSort: 4855 ms
Time of Hashtable + mergeSort: 4458 ms
```

Java - Project 2/src/ADTSortingTimer.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access | Java | Debug

Package ... | JUnit

- CorrelatorResults.j
- DataCount.java
- DataCounter.java
- DataCountStringC
- FileWordReader.jav
- FourHeap.java
- GArrayStack.java
- GStack.java
- Hasher.java
- Hashtable.java
- MoveToFrontList.j
- PriorityQueue.java
- SimpleIterator.java
- StringComparator.
- StringHasher.java
- StringHasher2.java
- StringHasher3.java
- TestAVLTree.java
- TestBinarySearchT
- TestFourHeap.java
- TestGArrayStack.ja
- TestHashtable.java
- TestMoveToFrontL
- TestSorting.java
- TestStringCompar
- WordCount.java
- WordCount2.java
- JRE System Library [JavaSE
- JUnit 4
- Test

WordCount.java | ADTSortingTimer.java | StringHasher2.java | StringHasher.java | StringHasher3.java | WordCount2.java

```
/**
 * Print out the time of processing the text and printing out the word and its count in descending order
 * used by different combinations of an abstract data type and a sorting algorithm.
 *
 * @author Chun-Wei Chen
 * @version CSE 332 02/19/13
 */
public class ADTSortingTimer {

    /**
     * @param args
     */
    public static void main(String[] args) {
        if (args.length != 1) {
            System.err.println("Invalid number of arguments");
            System.exit(1);
        } else {
            long startTime, endTime;
```

Javadoc | Declaration | Console | Debug

<terminated> ADTSortingTimer [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Feb 19, 2013 6:25:02 AM)

```
Timing Results:
Time of BinarySearchTree + insertionSort: 2759 ms
Time of BinarySearchTree + heapSort: 670 ms
Time of BinarySearchTree + mergeSort: 346 ms
Time of AVLTree + insertionSort: 87458 ms
Time of AVLTree + heapSort: 64364 ms
Time of AVLTree + mergeSort: 66527 ms
Time of MoveToFrontList + insertionSort: 4903 ms
Time of MoveToFrontList + heapSort: 6236 ms
Time of MoveToFrontList + mergeSort: 4358 ms
Time of Hashtable + insertionSort: 4589 ms
Time of Hashtable + heapSort: 4381 ms
Time of Hashtable + mergeSort: 3812 ms
```