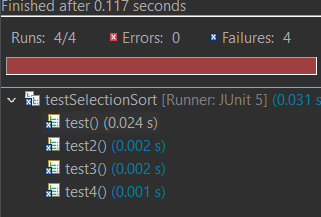
Lab 10

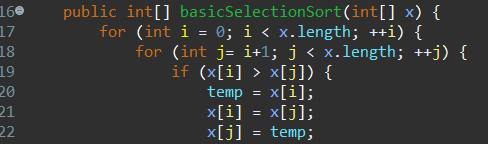
CIS 285

Sarah Eubank

Screenshots:

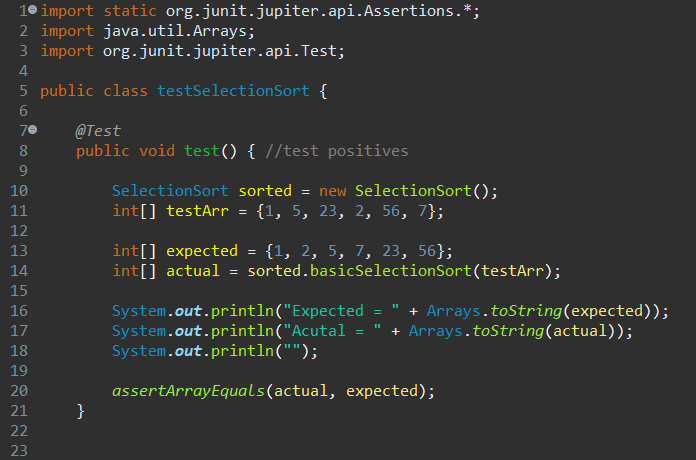
Part 1: (all code copied and pasted at end of report)



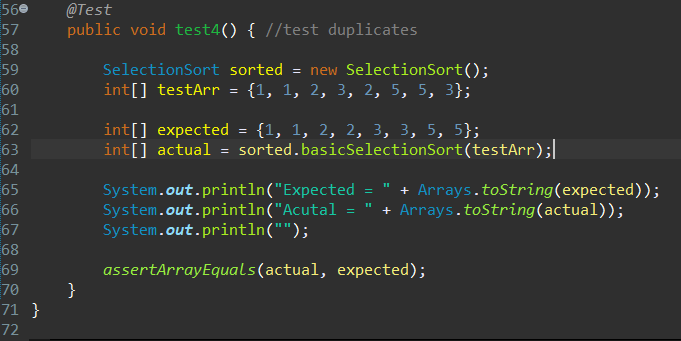


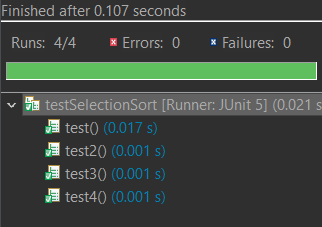
Changed the code so that it worked correctly by changing the last line in the nested for loop to x[j] = temp;

I didn’t see that some of the testing code was provided, so I created my own. Screenshots below.









Part 2:

Source Code:

*SelectionSort.java*

public class SelectionSort {

private int temp;

/\*\* Creates a new instance of SelectionSort \*/

public SelectionSort() {

}

/\* A simple SelectionSort algorithm

\* pre-condition:

\* post-condition:

\* inputs:

\* outputs:

\* special conditions:

\*/

public int[] basicSelectionSort(int[] x) {

for (int i = 0; i < x.length; ++i) {

for (int j= i+1; j < x.length; ++j) {

if (x[i] > x[j]) {

temp = x[i];

x[i] = x[j];

x[j] = temp;

}

} // end of inner for loop

} // end of outer for loop

return x;

} // end of basicSelectionSort method

}

*testSelectionSort.java*

import static org.junit.jupiter.api.Assertions.\*;

import java.util.Arrays;

import org.junit.jupiter.api.Test;

public class testSelectionSort {

*@Test*

public void test() { //test positives

SelectionSort sorted = new SelectionSort();

int[] testArr = {1, 5, 23, 2, 56, 7};

int[] expected = {1, 2, 5, 7, 23, 56};

int[] actual = sorted.basicSelectionSort(testArr);

System.***out***.println("Expected = " + Arrays.*toString*(expected));

System.***out***.println("Acutal = " + Arrays.*toString*(actual));

System.***out***.println("");

*assertArrayEquals*(actual, expected);

}

*@Test*

public void test2() { //test negatives

SelectionSort sorted = new SelectionSort();

int[] testArr = {-5, -4, -3, -1, -2};

int[] expected = {-5, -4, -3, -2, -1};

int[] actual = sorted.basicSelectionSort(testArr);

System.***out***.println("Expected = " + Arrays.*toString*(expected));

System.***out***.println("Acutal = " + Arrays.*toString*(actual));

System.***out***.println("");

*assertArrayEquals*(actual, expected);

}

*@Test*

public void test3() { //test positives and negatives

SelectionSort sorted = new SelectionSort();

int[] testArr = {1, -2, 3, 4, -5};

int[] expected = {-5, -2, 1, 3, 4};

int[]actual = sorted.basicSelectionSort(testArr);

System.***out***.println("Expected = " + Arrays.*toString*(expected));

System.***out***.println("Acutal = " + Arrays.*toString*(actual));

System.***out***.println("");

*assertArrayEquals*(actual, expected);

}

*@Test*

public void test4() { //test duplicates

SelectionSort sorted = new SelectionSort();

int[] testArr = {1, 1, 2, 3, 2, 5, 5, 3};

int[] expected = {1, 1, 2, 2, 3, 3, 5, 5};

int[] actual = sorted.basicSelectionSort(testArr);

System.***out***.println("Expected = " + Arrays.*toString*(expected));

System.***out***.println("Acutal = " + Arrays.*toString*(actual));

System.***out***.println("");

*assertArrayEquals*(actual, expected);

}

}