Name: \_Daniel Grimshaw\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Searching and Selection Sort Assignment**

Name your word document LastnameSelectionSort.docx

1. For a selection sort containing n elements, how many comparisons are required before the array is completely sorted? \_\_\_n squared\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Consider the following array of int elements.

int[] numbers = {7, 2, 8, 4, 1, 11, 9, 5, 3, 10};

Show the state of the elements after five passes of the outermost loop of the selection sort has occurred. Assume ascending order.

\_\_\_\_{1,2,3,4,5,11,9,7,8,10}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Suppose the following arrays below have been declared. For each array, write the state of the array after each pass of the outermost loop of the selection sort algorithm has occurred (after each element is selected and moved into place). Assume ascending order.
2. int[] numbers1 = {63, 9, 45, 72, 27, 18, 54, 36};

pass 1: \_\_\_\_{9,63,45,72,27,18,54,36}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pass 2: \_\_\_\_{9,18,45,72,27,63,54,36}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pass 3: \_\_\_\_{9,18,27,72,45,63,54,36}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. int[] numbers2 = {37, 29, 19, 48, 23, 55, 74, 12};

pass 1: \_{12,29,19,48,23,55,74,37}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pass 2: \_{12,19,29,48,23,55,74,37}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pass 3: \_\_\_{12,19,23,48,29,55,74,37}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. int[] numbers3 = {8, 5, -9, 14, 0, -1, -7, 3};

pass 1: \_\_{-9,5,8,14,0,-1,-7,3}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pass 2: \_\_{-9,-7,8,14,0,-1,5,3}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pass 3: \_\_{-9,-7,-1,14,0,8,5,3}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. int[] numbers4 = {15, 56, 24, 5, 39, -4, 27, 10};

pass 1: \_\_\_\_{-4, 56, 24, 5, 39, 15, 27, 10}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pass 2: \_\_\_\_{-4, 5, 24, 56, 39, 15, 27, 10}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pass 3: \_\_\_\_{-4, 5, 10, 56, 39, 15, 27, 24}\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. For the method below, write a comment before each statement explaining the purpose of the statement.

public void mystery() {

int pos, m; //declare pos and m

for (int i = 0; i < a.length - 1; i++) { //loop from 0 to one less than the length of a  
 m= a[i]; // set m to the element of a at i

pos = i; // set pos to i

for (int j = i + 1; j < a.length; j++) { //loop through the rest of the array

if (m < a[j]) { // if a[j] is greater than m

m = a[j]; // set m to the greatest

pos = j; // pos is updated

}

}

int temp = a[i]; //swap the element at i with the largest element

a[i] = a[pos];

a[pos] = temp;

}

}

1. What is the result from executing the following code compared with the code in the previous question? \_\_\_\_this will sort by increasing order.

public void mystery() {

int pos, m;

for (int i = 0; i < a.length - 1; i++) {  
 m = a[i];

pos = i;

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

if (a[j] < m) {

m = a[j];

pos = j;

}

}

int temp = a[i];

a[i] = a[pos];

a[pos] = temp;

}

}

1. What is the complexity for each of the following code segments?

a. \_\_\_O(n)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

for (int i = 1; i <= n; i++) {

System.out.println(i);

}

b. \_\_\_O(n squared)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

for (int i = 1; i <= n; i++) {

for (int j = 1; j <=n; j++) {

System.out.println(i\*j);

}

}

c. \_\_\_O(0)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int x = 5;

int y = 10;

int temp = x;

x = y;

y = temp;

7. Complete a. or b. below along with a main method in a file named LastnameMSelection.java if completing a. or LastnameDSelection.java if completing b. The main method needs to create and print the arrays specified below, then call one of the methods below for each array and then print each array again.

Prior to sorting:

array1: {22, 18, 12, -4, 27, 30, 36, 50, 7, 68, 91, 56, 2, 85, 42, 98, 25}

array2: {55}

array3: {98, 88, 76, 54, 45, 36, 32, 31, 31, 25, 19, 17, 12, 7, 5, 0, -10}

a. Write a modified version of the selection sort algorithm that selects the largest element each time and moves it to the end of the array, rather than selecting the smallest element and moving it to the beginning. It still sorts the array in ascending order.

b. Write a modified “dual” version of the selection sort algorithm that selects both the largest and smallest elements on each pass and moves each of them to the appropriate end of the array. It still sorts the array in ascending order.