### Building Java Programs Chapter 13 Lab Handout

### Binary Search

1. Suppose that that we have an array called list initialized as follows:

int[] list = {-2, 8, 13, 22, 25, 25, 38, 42, 51, 103};

This would construct the following array:

[0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

| -7 | 8 | 13 | 22 | 25 | 25 | 38 | 42 | 51 | 100 |

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

a) What values would low, high and mid take on for the following call:

binarySearch(list, 23)

and what value would be returned?

b) What values would low, high and mid take on for the following call:

binarySearch(list, 2, 8, 38) Note: Use the Author’s code for this question

and what value would be returned?

### Efficiency and Big-Oh Notation

2. Approximate the value of sum after the following code fragment, in terms of variable *n* in Big-Oh notation.

int sum = 0;

for (int i = 1; i <= n - 3; i++) {

for (int j = 1; j <= n + 4; j += 5) {

for (int k = 1; k <= n + 4; k += 5) {

sum += 2;

}

}

sum++;

}

for (int i = 1; i <= 100; i++) {

sum++;

}

3. Approximate the value of sum after the following code fragment, in terms of variable *n* in Big-Oh notation.

int sum = 0;

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

sum++;

}

for (int j = 1; j <= n / 2; j++) {

for (int k = 1; k <= n / 2; k++) {

sum++;  
 }

}

### Sorting

4. Consider the following array of int values.

[17, 11, 6, 22, -3, 8, 4, 22, 2, 35, -4, 10]

1. Write the contents of the array after 4 passes of the outermost loop of **selection sort**.  
    Assume selection sort is sorting slots from low-to-high
2. Write the contents of the array during each of the the recursive calls of **merge sort**.

### Chapter 13 Lab Handout Solutions

1.

The variables would take on the following sequence of values:

low high mid

-------------------

0 9 4

0 3 1

2 3 2  
 3 3 3

and would return the value -5.

b) The variables would take on the following sequence of values:

low high mid

-------------------

2 8 5

6 8 7

6 6 6

and would return the value 6.

2. O(n3)

3. O(n2)

4.

selection sort, first 4 passes (sorting slots low to high)

1 [-4, 11, 6, 22, -3, 8, 4, 22, 2, 35, 17, 10]

2 [-4, -3, 6, 22, 11, 8, 4, 22, 2, 35, 17, 10]

3 [-4, -3, 2, 22, 11, 8, 4, 22, 6, 35, 17, 10]

4 [-4, -3, 2, 4, 11, 8, 22, 22, 6, 35, 17, 10]

merge sort

[17, 11, 6, 22, -3, 8, 4, 22, 2, 35, -4, 10]

[17, 11, 6, 22, -3, 8]

[ 17, 11, 6]

[ 17]

[ 11, 6]

[ 11]

[ 6]

[ 6, 11]

[ 6, 11, 17]

[22, -3, 8]

[22]

[-3, 8]

[-3]

[ 8]

[-3, 8]

[-3, 8, 22]

[-3, 6, 8, 11, 17, 22]

[ 4, 22, 2, 35, -4, 10]

[ 4, 22, 2]

[ 4]

[22, 2]

[22]

[ 2]

[ 2, 22]

[ 2, 4, 22]

[35, -4, 10]

[35]

[-4, 10]

[-4]

[ 10]

[-4, 10]

[-4, 10, 35]

[-4, 2, 4, 10, 22, 35]

[-4, -3, 2, 4, 6, 8, 10, 11, 17, 22, 22, 35]