Programming Fundamentals: Lab 6

**Exercise 2:** Find the time complexity of the following Java method:

int foo(int N) {

int result = 0; // O(1)

for (int i = 0; i < N; i++) // O(N)

result++; // O(1)

for (int j = 0; j < 1000000; j++) // O(1)

result += j; // O(1)

return result; // O(1)

}

The for loop iterating from i to N has the largest time complexity as all other portions of the method are constant. Therefore, the overall time complexity for this method is O(N).

**Exercise 3:** Find the time complexity of the following Java method:

int bar(int N) {

int result = 1; // O(1)

for (int i = 1; i < N; i \*= 2) // O(log N)

result += 2; // O(1)

return result; // O(1)

}

The for loop has the largest time complexity. Since i is incrementing by multiples of 2, the time complexity is O(log N).

**Exercise 4:** Assume a binary search is performed on the following array of integers:

{1, 14, 15, 24, 55, 59, 73, 90, 94, 99}

Trace through each iteration of the algorithm, writing the number that will be the middle element and the left and right bounds (indexes), when searching for the number 73.

The array has an even number of elements, so a choice can be made when selecting the middle value. The search will be different depending on whether the left or right midpoint is chosen.

Selecting the left element of the midpoint:

1st iteration: The left index is 0 and the right index is 9 yielding the middle value of 55 (index = 4).

{1, 14, 15, 24, **55**, 59, 73, 90, 94, 99}

2nd iteration: We are searching for the value 73 which is larger than 55 so we need to search to the right. The left index is 5 and the right index is 9, with a middle value of 90 (index = 7).

{1, 14, 15, 24, 55, 59, 73, **90**, 94, 99}

3rd iteration: The value 73 is less than 90 so the search will happen to the left. The left index is 5 and the right index is 6 with a middle value of 59 (index = 5).

{1, 14, 15, 24, 55, **59**, 73, 90, 94, 99}

4th iteration: The value 73 is larger than 59 so the search will occur to the right. The left index is 6 and the right index is 6, with the middle value of 73, which is the number we are looking for so the binary search terminates. {1, 14, 15, 24, 55, 59, **73**, 90, 94, 99}

Selecting right element of the midpoint:

1st iteration: The left index is 0 and the right index is 9 yielding the middle value of 59 (index = 5).

{1, 14, 15, 24, 55, **59**, 73, 90, 94, 99}

2nd iteration: We are searching for the value 73 which is larger than 59 so we need to search to the right. The left index is 6 and the right index is 9, with a middle value of 94 (index = 8).

{1, 14, 15, 24, 55, 59, 73, 90, **94**, 99}

3rd iteration: The value 73 is less than 94 so the search will happen to the left. The left index is 6 and the right index is 7 with a middle value of 90 (index = 7).

{1, 14, 15, 24, 55, 59, 73, **90**, 94, 99}

4th iteration: The value 73 is less than 90 so the search will occur to the left. The left index is 6 and the right index is 6, with the middle value of 73, which is the number we are looking for, so the binary search terminates. {1, 14, 15, 24, 55, 59, **73**, 90, 94, 99}

**Exercise 5:** Trace the execution of the insertion and selection sort algorithms when executed on the following array of integers: {1, 29, 14, 15, 94}

Show how the array will look after each iteration of the outer loop.

Insertion

1st iteration: {1, 29, 14, 15, 94} 1 is sorted. 29 is larger so nothing is shifted and 29 is inserted.

2nd iteration: {1, 29, 14, 15, 94} 1 and 29 are sorted. 29 is shifted and 14 is inserted.

3rd iteration: {1, 14, 29, 15, 94} 1, 14, and 29 are sorted. 29 is shifted and 15 is inserted.

4th iteration: {1, 14, 15, 29, 94} 1, 14, 15, and 29 are sorted. 94 is larger so nothing is shifted and the array is sorted.

Selection

1st iteration: The array is scanned to the right starting at 1 (index = 0). The smallest element is 1 so nothing is exchanged. {1, 29, 14, 15, 94}



2nd iteration: The array is scanned to the right starting at 29 (index = 1). The smallest element is 14 so 14 and 29 are exchanged. {1, 29, 14, 15, 94}



3rd iteration: The array is scanned to the right starting at 29 (index = 2). The smallest element is 15 so 29 and 15 are exchanged. {1, 14, 29, 15, 94}



4th iteration: The array is scanned to the right starting at 29 (index = 3). The smallest element is 29 so 29is exchanged with 29. {1, 14, 15, 29, 94}

