**Time of cost**

number1 <- String input converted to Int 1

number2 <- String input converted to Int 1

List <- [4, 19, 23, 36, 40, 43, 61, 64, 78, 95] 1

binarySearch(num1, num2, array) {

midpoint <- len(array)//2 1

If num1 <= num2 { 1

If array[midpoint] ≥ num2 AND array[midpoint] ≤ num2 { 1

Return TRUE 1

}

ElseIf midpoint > num2 { 1

Return binarySearch(num1, num2, right half of array) n

}

ElseIf midpoint < num2 { 1

Return binarySearch(num1, num2, left half of array) n

Return FALSE 1

}

Else {

Return “Error! Lower value > upper value” 1

}

}

**Runtime bound**: f(n) = 2n + 11

**Complexity:** O(n)

**AFTER CONSIDERING OF THE EFFECT RECURSION**

**Time of cost**

number1 <- String input converted to Int 1

number2 <- String input converted to Int 1

List <- [4, 19, 23, 36, 40, 43, 61, 64, 78, 95] 1

binarySearch(num1, num2, array) {

midpoint <- len(array)//2 n

If num1 <= num2 { n

If array[midpoint] ≥ num2 AND array[midpoint] ≤ num2 { n

Return TRUE 1

}

ElseIf midpoint > num2 { n

Return binarySearch(num1, num2, right half of array) n

}

ElseIf midpoint < num2 { n

Return binarySearch(num1, num2, left half of array) n

Return FALSE 1

}

Else {

Return “Error! Lower value > upper value” 1

}

}

**Runtime bound**: f(n) = 7n + 6

**Complexity:** O(n)