ArrayUtil Steps

**ArrayUtil Class**

**intArray Field:** Integer array field intArray

**Default constructor:** public ArrayUtil

**Constructor with one argument:**

Make a constructor with a parameter of arrayUtility and set intArray to that parameter

**Accessor:**

Getter method

Gets value within intArray

**Mutator:**

Setter method

Changes value within intArray using the parameters that give the proper index and value

**minValue:**

if statement that checks if the length of intArray is 0 and if so, it returns 0

temporarySmallest variable declaration and initialized to the first value

for loop that starts at index 1 since temporarySmallest is already index 0 and goes until right before intArray’s length

Compare temporarySmallest (first value) to second value

Store the smaller value in a variable (temporarySmallest)

Compare (new) temporarySmallest to the third value

Store the smaller value in a variable (temporarySmallest)

Keep going until the end and return temporarySmallest

**maxValue:**

if statement that checks if the length of intArray is 0 and if so, it returns 0

temporaryBiggest variable declaration and initialized to the first value

for loop that starts at index 1 since temporarySmallest is already index 0 and goes until right before intArray’s length

Compare temporaryBiggest (first value) to second value

Store the bigger value in a variable (temporaryBiggest)

Compare (new) temporaryBiggest to the third value

Store the bigger value in a variable (temporaryBiggest)

Keep going until the end and return temporaryBiggest

**countUniqueIntegers:**

if statement that checks if the length of intArray is 0 and if so, it returns 0

uniqueCounter variable declared and initialized to 0

uniqueState variable declared and initialized to true (this will act as a Boolean flag so we can check whether we are able to say if it was unique and add 1 to the uniqueCounter)

Outer for loop that is for each of the digits we want to check (0 to intArray’s length)

Inner for loop for comparing the outer for loop’s value to the rest (1 to intArray’s length)

If statement inside inner for loop to check if outer for loop’s value does not equal inner for loop’s value

If so, uniqueState continues to be true and continues the for loop

If not, uniqueState becomes false and the inner for loop breaks

Before the inner for loop ends, another if statement checks if uniqueState is still true

If so, add 1 to unique counter

Outside of the inner for loop, uniqueState is set back to true in order to continue the other for loops’ cycle

Outside of the outer for loop, return unique counter