**1a)**

**1b)**

**1c)**

**1d)**

**1e)**

**2a)** The worst-case running time for this algorithm with an array of size n is O(n). The algorithm iterates over the array and doesn’t perform any operations that take up significant time. This worst case involves *val* being at the end of *data*:

In this case, the entirety of *data* must be iterated over before *val* is found, resulting in the worst-case running time of O(n).

**2b)** The best-case running time for this algorithm with an array of size n is O(1). This best-case scenario requires *val* to be at the start of *data*:

Any array of this type will result in a running time of O(1), as *val* is the first element checked.

**3)** Below are all the required methods (with code), the explanation behind their O(1) runtime, and an example test showing the function of *forget(int k).*

**void push()**

**A screen shot of a computer code

AI-generated content may be incorrect.**

This push() method is O(1) as there are no operations being done based on the stack’s size. O(1)

**E pop()**

**A screen shot of a computer code

AI-generated content may be incorrect.**

This pop() method is O(1) for the same reasoning as the previous push() method. There are only pointer assignment operations being performed on the stack. O(1)

**void forget(int k)**

**A screen shot of a computer

AI-generated content may be incorrect.**

While this may seem O(n) at first with the for() loop, the loop doesn’t run off the stack’s size and is therefore constant for all stack sizes when comparing them at the same k value. O(1)

**int size()**

****

This just returns the size variable. O(1)

**boolean isEmpty()**

****

This returns true if the size variable is 0, false otherwise. O(1)

**E top()A computer screen shot of text

AI-generated content may be incorrect.**

This returns the top node’s element, assuming top isn’t null. If it is, the method just throws an EmptyStackException. O(1)

**Testing**

Below is the main method (included in the attached java file) with test code along with its console output. (The toString() method I used is in the attached java file)

A computer screen shot of a program

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.

In the output image above, the top() method at the end is called 4th instead of second due to it throwing an exception when a ForgetfulStack<E> is empty