

# EECS3101 Assignment 1

Jerry Wu (217545898)

Due Feb. 3 2023 at 22:00

## Question 1

Precondition:: The array  $lst$  contains only characters ' $A$ ', ' $B$ ', ' $C$ '.

Postcondition::  $lst$  is sorted alphabetically in non descending order.

- a) Our job is to find  $Inv(p_1, p_2, p_3)$  of the while loop given in the code.

In the code, it can be observed that  $p_3$  is the variable that gets incremented at the end of each iteration regardless of which branch the loop goes to.

Because  $p_3$  is in the loop guard, we can label it as the traversal variable, and both  $p_1$  and  $p_2$  are case variables for characters ' $A$ ' and ' $B$ ' respectively.

We can traverse this algorithm using a sample input list. Let us define the following arbitrary list to find a pattern of some kind::

$$lst = [B, B, A, A, C, B, C, A]$$

We can now trace the algorithm.

**1st pass:**  $p_1 = 0, p_2 = 1, p_3 = 1$

**2nd pass:**  $p_1 = 0, p_2 = 2, p_3 = 2$

**3rd pass:**  $p_1 = 1, p_2 = 3, p_3 = 3$

**4th pass:**  $p_1 = 2, p_2 = 4, p_3 = 4$

**5th pass:**  $p_1 = 2, p_2 = 4, p_3 = 5$

**6th pass:**  $p_1 = 2, p_2 = 5, p_3 = 6$

**7th pass:**  $p_1 = 2, p_2 = 5, p_3 = 7$

**8th pass:**  $p_1 = 3, p_2 = 6, p_3 = 8$

Observing this output from the sample input, we notice that  $p_3$  is always greater than or equal to  $p_2$  and  $p_1$ .

Thus our invariant can be written as the following predicate::

$$Inv(p_1, p_2, p_3) \equiv (p_3 \geq p_2 \geq p_1)$$

- b) Now, we can begin our proof of the invariant by **cases**. For this invariant, we have two cases that we need to invoke induction on in order to prove it completely. For the sake of clarity, let us define  $lst.length = n$ . Thus our loop guard can be rewritten as  $E \equiv p_3 < n$ .

**Lemma 1::** Because  $p_3 < n$  is assumed to be true, it would follow that  $p_2 < n$  and  $p_1 < n$  are also true.

### Case 1:: $p_3 \geq p_2$

Our job is to show that each 'B' in the array precedes every 'C' in the array.

### Proof of case 1::

**Abstract::** Define  $p_{3b}$  and  $p_{2b}$  to be the values of  $p_3$  and  $p_2$  before each iteration, and  $p_{3a}$  and  $p_{2a}$  for after each iteration. We assume that  $p_{3b} \geq p_{2b}$ . Our job is to show that  $p_{3a} \geq p_{2a}$ .

**Pf::** Upon inspection, it's evident that  $p_{3a} = p_{3b} + 1$ , and  $p_{2a} = p_{2b} + 1$  after each iteration. However, it is obviously the case that  $p_{2a} = p_{2b} + 1$  will only have a 1 in 3 chance to occur. This is because  $p_{2a} = p_{2b} + 1$  is attached to a condition, whereas  $p_{3a} = p_{3b} + 1$  happens absolutely with no condition attached. This would in turn lead one to believe that under any circumstances given the conditional bound of  $p_{2a} = p_{2b} + 1$ , that  $p_{3a} = p_{3b} + 1$  occurs more times overall, or an equal number of times that  $p_{2a} = p_{2b} + 1$  occurs, implying that  $p_3 \geq p_2$ . Hence, case 1 is proved.

As a consequence of this, it has also been shown that  $p_3 \geq p_1$  because  $p_1$  being incremented is also bound by a condition.

### Case 2:: $p_2 \geq p_1$

Next, let us show that every 'A' in the array precedes every 'B' in the array. We can do this simply by inspection. In the code, we can see that in the first if condition,  $p_2 = \max\{p_2, p_1\}$ , leading one to believe that  $p_2$  will indeed always be a larger number than  $p_1$ . Thus proves that each 'A' precedes each 'B'.

### Case 3:: $p_3 \geq p_1$

Because we already showed that  $p_3 \geq p_2$  and  $p_2 \geq p_1$ , it would suffice to say that  $p_3 \geq p_1$  because of the above two corollaries, that each 'A' in the array precedes each 'C' in the array.

Thus our invariant  $Inv(p_1, p_2, p_3) \equiv (p_3 \geq p_2 \geq p_1)$  has been proved. ■

- c) The postcondition being proven was a consequence of our proof for the invariant in part b) due to how the invariant itself tells us that the letters will become ordered. This is because each case shows that each corresponding letter in the array will **always** precede the one coming after it (i.e.  $p_3 \geq p_2$  and  $p_2 \geq p_1$ ).
- d) We can say for certain beyond reasonable doubt that this loop will terminate. The loop guard is  $p_3 < lst.length$ . We know the value of  $p_3$  will eventually equal  $lst.length$  because it is being incremented once at the end of every iteration, and this operation is not tied to any if/else conditions. From this proposition it would follow, that the loop as a whole is complete and correct.

## Question 2

Precondition::  $a, b \in \mathbb{N}$

Postcondition:: return  $LCM(a, b)$

- a) Our job is to find the invariant for the loop in this function. We can start by tracing the algorithm with example input (5,7).

**1st pass:**  $x = 5, y = 7$

**2nd pass:**  $x = 10, y = 7$

**3rd pass:**  $x = 10, y = 14$

**4th pass:**  $x = 15, y = 14$

**5th pass:**  $x = 15, y = 21$

**6th pass:**  $x = 20, y = 21$

**7th pass:**  $x = 25, y = 28$

**8th pass:**  $x = 30, y = 28$

**9th pass:**  $x = 30, y = 35$

**10th pass:**  $x = 35, y = 35$

So the final result of this run would be that  $LCM(5, 7) = 35$

Upon inspection, clearly we can see that  $Inv(x, y) \equiv (x \leq LCM(a, b)) \wedge (x > 0) \wedge (y > 0)$ .