CSC148 Worksheet 2 Solution

Hyungmo Gu

April 16, 2020

Question 1

Relevant Property	Values to Try
The position of $< n1 >$ in lst	front, back, somewhere else
The position of $< n1 >$ after insertion	front, back, and somewhere else
The position of $< n2 >$ beside $< n1 >$ after insertion	front, back and somewhere else
Size of lst	- Size of lst after insertion - Size of list before insertion

Correct Solution:

Relevant Property	Values to Try
The position of $< n1 >$ in lst	front, back, somewhere else
Length of list	0,1,'small' value
Number of occurences of lst	0,1,'small' value,every value in lst is $< n1 >$
$< n1 > == < n2 >$	true, false

Question 2

-

lst	n1	n2	Purpose
$[0, 1, 2, 3]$	0	99	$n1$ at the front
$[0, 1, 2, 3]$	0	99	$n1$ at the back
$[0, 1, 2, 3]$	3	99	$n1$ at somewhere else
$[0, 1, 2, 3]$	3	3	$< n1 >$ the same as $< n2 >$
$[0, 1, 2, 3]$	3	4	$< n1 >$ not the same as $< n2 >$
$[]$	3	4	list with length of 0
$[1]$	3	4	list with length of 1
$[1, 2, 3, 5, 6]$	3	4	list with length of 'small' value
$[1, 5, 6, 7]$	3	4	list with 0 occurrences of $< n1 >$
$[1, 3, 5, 6]$	3	4	list with 1 occurrences of $< n1 >$
$[3, 3, 3, 3]$	3	4	list with every occurrences of $< n1 >$

Correct Solution:

lst	n1	n2	Purpose
$[0, 1, 2, 3]$	0	99	$n1$ at the front
$[0, 1, 2, 3]$	3	99	$n1$ at the back
$[0, 1, 2, 3]$	1	99	$n1$ at somewhere else
$[0, 1, 2, 3]$	3	3	$< n1 >$ the same as $< n2 >$
$[0, 1, 2, 3]$	3	4	$< n1 >$ not the same as $< n2 >$
$[]$	3	4	list with length of 0
$[1]$	3	4	list with length of 1
$[1, 2, 3, 5, 6]$	3	4	list with length of 'small' value
$[1, 5, 6, 7]$	3	4	list with 0 occurrences of $< n1 >$
$[1, 3, 5, 6]$	3	4	list with 1 occurrences of $< n1 >$
$[3, 3, 3, 3]$	3	4	list with every occurrences of $< n1 >$

Question 3

- Test for '$n1$ at the back'

```python
    def test_insert_after_at_back() -> None:
        """Test insert_after with one occurrence of n1 at the back of lst.
        """

        input_list = [0,1,2,3]
        input_after(input_list, 3, 99)
        expected = [0,1,2,3,99]

        assert input_list == expected

```

- Test for ‘$n1$ at the somewhere else’

```python
    def test_insert_after_somewhere_else() -> None:
        """Test insert_after with one occurrence of n1 at somewhere
    else in lst.
        """

        input_list = [0,1,2,3]
        input_after(input_list, 3, 99)
        expected = [0,1,99,2,3]

        assert input_list == expected

```

- Test for ‘$n1$ the same as $n2$’

```python
    def test_insert_after_somewhere_else() -> None:
        """Test insert_after with <n1> the same as <n2>
        """

        input_list = [0,1,2,3]
        expected = [0,1,1,2,3]
        insert_after(input_list, 1,1)

        assert input_list == expected

```

- Test for ‘$n1$ not the same as $n2$’

```python
    def test_insert_after_somewhere_else() -> None:
        """Test insert_after with <n1> not the same as <n2>
        """

        input_list = [0,1,2,3]
        expected = [0,1,3,2,3]
        insert_after(input_list, 1,3)

        assert input_list == expected

```

- Test for ‘list with length of 0’

```python
    def test_insert_after_somewhere_else() -> None:
        """Test insert_after with list with length of 0
        """

        input_list = []
        expected = []
        insert_after(input_list, 3,4)

        assert input_list == expected

```

- Test for ‘list with length of 1’

```python
    def test_insert_after_somewhere_else() -> None:
        """Test insert_after with list with length of 1
        """

        input_list = [1]
        expected = [1]
        insert_after(input_list, 3,4)

        assert input_list == expected

```

- Test for ‘list with length of small value’

```python
    def test_insert_after_somewhere_else() -> None:
        """Test insert_after with list with length of small value
        """

        input_list = [1,2,3,5,6]
        expected = [1,2,3,4,5,6]
        insert_after(input_list, 3,4)

        assert input_list == expected

```

- Test for ‘list with 0 occurrence of $n1$’

```python
    def test_insert_after_somewhere_else() -> None:
        """Test insert_after with list with 0 occurrence of n1
        """

        input_list = [1,5,6,7]
        expected = [1,5,6,7]
        insert_after(input_list, 3,4)

        assert input_list == expected

```

- Test for ‘list with 1 occurrence of $n1$’

```python
    def test_insert_after_somewhere_else() -> None:
        """Test insert_after with list with 1 occurrence of n1
        """

        input_list = [1,3,5,6]
        expected = [1,3,4,5,6]
        insert_after(input_list, 3,4)

        assert input_list == expected

```

- Test for ‘list with every occurrence of $n1$’

```python
    def test_insert_after_somewhere_else() -> None:
        """Test insert_after with list with 1 occurrence of n1
        """

        input_list = [3,3,3,3]
        expected = [3,4,3,4,3,4,3,4]
        insert_after(input_list, 3,4)

        assert input_list == expected

```