CSC148 Worksheet 8 Solution

Hyungmo Gu

April 21, 2020

Question 1

- No. It's not a good solution.

  The code is trying to count the number of elements in list.

  The *for* loop takes $\Theta(n)$ time, and this is not an efficient solution.

  We can do better than that by reducing the runtime to $\Theta(1)$ by using *len(...)* function.

  Correct Solution:

  No. It's not a good solution.

  Stacks are not iterable

Question 2

- Yes. This is a good solution.

  The quick points are

  - The method is trying to determine the number of elements in *Stack*.
  - *pop()* method removes an element from stack. This works as an indexing variable for the while loop.
  - *is_empty()* method checks for the condition of stack not having any elements. This allows while loop to terminate after using stack's *pop()* method sufficient number of times.
  - *count* variable allows the number of elements to be counted, as it is being removed from *Stack* by *pop* method.

Correct Solution:

No. This is not a good solution.

The quick points are

- The code uses *pop()* method.
- *pop()* method causes *Stack* to mutate in number of elements, and the next time the *size* function is called, it will return 0.
- *size()* function should not affect the number of elements in stack.

Question 3

- This is a good solution if the instance attribute *_items* is using list to store items.

  Going further, this is a good solution for any iterable objects with *__len__* method (it should be correctly defined as well!).

Correct Solution:

No. This is not a good solution.

s._item is a private attribute, and private attribute should not be used outside of *Stack*.

Question 4

- No. This is not a good solution.

  The quick points are

  - Parameter *s* is of type *Stack*
  - *Stack* is a class
  - Class passes function by reference. That is, changes made to class inside function also affects outside.
  - The variable *s_copy* is pointing to *s*
    - Unlike with string and integers, this doesn't copy class (this is a huge bad)
  - *pop()* method is used, and this causes *s* outside of function to have size 0 by the end of operation

Question 5

```
from typing import Any

class Stack:
    """A last-in-first-out (LIFO) stack of items.
    Stores data in last-in, first-out order. When removing an item
from the
    stack, the most recently-added item is the one that is removed.
    """
    def __init__(self) -> None:
        """Initialize a new empty stack."""
        self._item = []

    def is_empty(self) -> bool:
        """Return whether this stack contains no items.
        >>> s = Stack()
        >>> s.is_empty()
        True
        >>> s.push('hello')
        >>> s.is_empty()
        False
        """

        if len(self._item) != 0:
            return False

        return True

    def push(self, item: Any) -> None:
        """Add a new element to the top of this stack.
        """
        self._item.append(item)

    def pop(self) -> Any:
        """Remove and return the element at the top of this stack.
        >>> s = Stack()
        >>> s.push('hello')
        >>> s.push('goodbye')
        >>> s.pop()
        'goodbye'
        """
        if len(self._item) == 0:
            return None

        item = self._item.pop()
        return item

if __name__ == '__main__':
    import doctest
    doctest.testmod()
```

Listing 1: worksheet_8_solution.py