Brian Huang
CS261

Homework 3 written questions

1. To create a function that finds the size of a linked stack that has no fields I would:
    a. Create a copy of the stack.
    b. Create a counter variable.
    c. Pop the stack until it's empty. While I pop the stack I would increment the counter variable.
    d. Then I would return the counter variable.

   This function has a run time complexity of O(n^2).

2. I fields were allowed for the implementation I would pass the stack into the function. Instead of copying I would just traverse the stack in a while loop until I hit the NULL pointer while incrementing a counter. Traversing would look like:
    a. traverse = stack -> next

   This function would have a run time complexity of O(n).

3. To compute the size of a stack implemented in an array I would just have the function return the size of the array.
    a. Return array_stack -> size

   The run time complexity of this is just O(1).

4. To implement a stack by using a deque I would just remove all the functions that relate to the back of the deque. Functions such as remove_back, add_back, check_back would never be used. Only functions related to the front would be used.

5. You can keep track of front and back by setting their value to NULL. Then for any operation concerning the front or the back you can just check if the node's value is NULL, then you will know you either reached the front or the back.