COS226 Week 2 Activity

1. *Design an algorithm.* Design a quadratic-time algorithm for the 3-sum problem. Describe your design by giving a crisp and concise English description of your algorithm; don't write Java code.

   (a) Given an integer x and a sorted array a[] of N distinct integers, design a linear-time algorithm to find if there exists indices i and j such that `(a[i] + a[j] == x)`. Hint: start by checking whether `a[0] + a[N-1] is <, >, or == x`.

   (b) Given an array a[] of N distinct integers, design a quadratic-time algorithm to find if there exists indices i, j, and k such that `(a[i] + a[j] + a[k] == 0)`. Hint: Use the result from (a). You can assume the array is sorted since sorting the array can be done in quadratic (and even linearithmic) time.

2. *Generics, iterators, autoboxing, and mathematical analysis.*

   Use the class shown here: http://algs4.cs.princeton.edu/13stacks/ResizingArrayStack.java.html

```
ResizingArrayStack<Character> stack = new ResizingArrayStack<Character>();
stack.push('A');
stack.push('C');
stack.push('T');
for (char left : stack)
    for (char right : stack)
         StdOut.println(left + " " + right);
```

   (a) What does the above code fragment output to standard output?

   (b) If N characters are pushed onto the stack initially (instead of 3), how many lines of output does the above code fragment produce as a function of N?

3. *Unit testing.* Describe three ways to test the correctness of the implementation of `ResizingArrayStack`.

   -
   -
   -

4. *Memory analysis.* Suppose you have a generic stack implemented using a linked list, as defined below:

```
public class Stack<Item> implements Iterable<Item> {
    private int N;          // size of the stack
    private Node first;     // top of stack

    private class Node {
        private Item item;
        private Node next;
    }
    ...
}
```

   (a) How much memory (in bytes) does a single `Node` object use? Use the 64-bit memory cost model from Section 1.4. Do not include the memory for the item itself—this memory is allocated by the client and depends on the item type.

   (b) How much memory (in bytes) does a `Stack` use to store N items? Do not include the memory for the items themselves.

   (c) Repeat the previous question, but use tilde notation to simplify your answer.