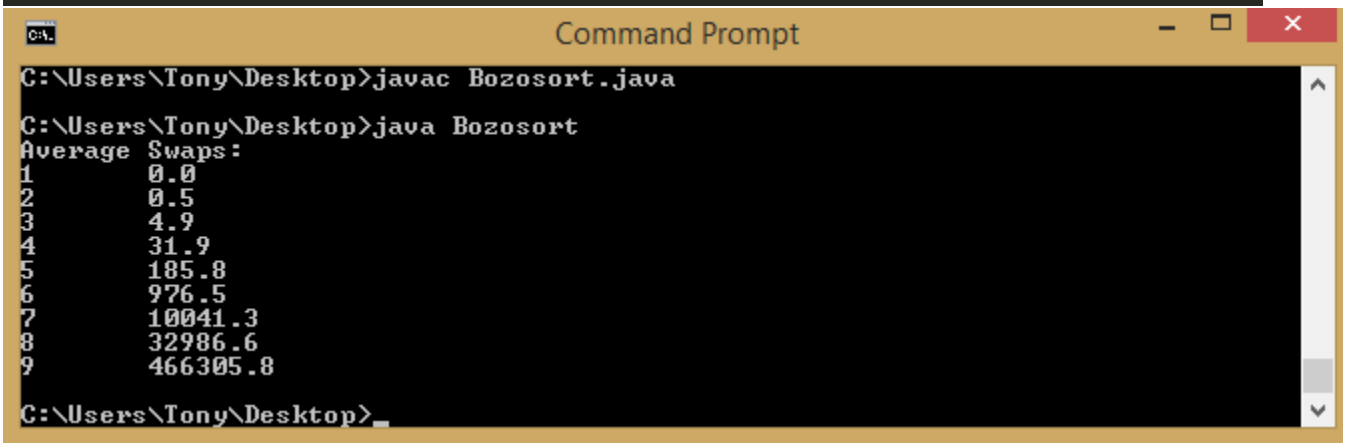
Homework 3

Anthony Menjivar

1 Java Bozosort

```java
import java.util.Random;

public class Bozosort {

    public static int[] randomizeArray(int size, Random randomize) {
        int[] intArray = new int[size];
        for (int i = 0; i < size; i++) {
            intArray[i] = randomize.nextInt();
        }
        return intArray;
    }

    private static boolean isSorted(int[] array) {
        for (int i = 1; i < array.length; i++) {
            if (array[i-1] > array[i]) {
                return false;
            }
        }
        return true;
    }

    public static int sort(int[] array, Random randomize) {
        int counter = 0;
        while (!isSorted(array)) {
            int int1 = randomize.nextInt(array.length);
            int int2 = randomize.nextInt(array.length);
            if (int1 != int2) {
                int intArray1 = array[int1];
                int intArray2 = array[int2];
                array[int1] = intArray2;
                array[int2] = intArray1;
                counter++;
            }
        }
        return counter;
    }

```

```java
    public static void main(String[] args) {
        Random random = new Random();
        int iterate = 10;
        System.out.println("Average Swaps:");
        for(int i = 1; i < 10; i++) {
            int swaps = 0;
            for (int k = 0; k < iterate; k++) {
                swaps += sort(randomizeArray(i, random), random);
            }
            System.out.println(i + "    " +swaps / (double)iterate);
        }
    }

}
```



2 Java Autokey Vigenere Cipher

```java
public class AutoKeyVigenere {

    public static String encipher(String text, String key) {
        return encipher(text, key, 1);
    }

    public static String decipher(String cipher, String key) {
        return encipher(cipher, key, -1);
    }

    private static String encipher(String text, String key, int multiply) {
        StringBuilder cipher = new StringBuilder();
        for (int i = 0, n = text.length(); i < n; i++) {
            char k = i < key.length() ? key.charAt(i) : multiply == 1 ? text.charAt(i - key.length()) : cipher.charAt(i - key.length());
            cipher.append((char)(k * multiply + text.charAt(i)));
        }
        String cipherText = cipher.toString();
        return cipherText;
    }
}
```

3 Monoalphabetic Substitution Cipher Plaintext

UIQLDEVORHIWLTQTOKMQMWROUOQQMQLKIQWQVIEWRDQTLEQMWRWXFTWHTOA
DMRDQIOKWXMAOHMRMRHQVOQWLTAOMRQODPMDQWMRDQTLEQOEWAFLQITBVOQ
QWKWUIQLDEWREIRQTOQITOQVITWRIJFUOMRMRHQWVLAORSIMRHDBVOQBIBO
RQOEWAFLQITQWKW

LETUSCHANGEOURTRADITIONALATTITUDETOTHECONSTRUCTIONOFPROGRAM
SINSTEADOFIMAGININGTHATOURMAINTASKISTOINSTRUCTACOMPUTERWHAT
TODOLETUSCONCENTRATERATHERONEXPLAININGTOHUMANBEINGSWHATWEWA
NTACOMPUTERTODO

4 bifid Algorithm Decryption

TWBTLLAEPODTUBTWBTLTDLDDVSNNHEETLSKDDSIFGIIMWLYDKDDSPHBPQKOFHMDLSKRS

COMPUTERSCIENCEISNOMOREABOUTCOMPUTERSTHANASTRONOMYISABOUTTELESCOPESS

5 Private Key

If someone's RSA public key is (729880581317, 5), what is her private key?

private key is (N, d)
$N = 729880581317$ $N = pq$
$p = 822893$ $q = 886969$
$(p-1)(q-1) = 729878871456$
$d = 5 InverseMod(729878871456)$
Private Key = $(729880581317, 583903097165)$

6 Problem 1.45

6.1 a

We would want digital signature schemes because they can be used to authenticate the sender's identity and to make sure that the message has not been altered by a third party during communication.

6.2 b

$((N, e), M^d, M)$
$(M^d)^e \bmod N = (M^e)^d \bmod N = M \bmod N = M$

If someone were able to sign given only (N, e), they would be able to exponentiate by $d \bmod N$, which allows decryption, contradicting RSA security. So if RSA is secure, this scheme is also secure.

6.3 c

$p = 137$ $q = 71$ $e = 99$
$N = pq = 9727$ and m_1^d
Extended Euclid:
$d = 6539$

6.4 d

RSA key = (17, 391)
Factor 391 to $17 * 23$
$\Phi(391) = 16 * 22 = 352$
Using Extended Euclid: $d = (e)^{-1}$ *mod* $352 = 145$

Check: $145 * 17 = 2465 = 1$ *mod* 352

7 Problem 1.46

7.1 a

The message sent by Alice to Bob is the encrypted message $M^e \ mod \ N$. With this, if Bob agreed to sign anything he is asked to with his private key, Eve will obtain $(M^e)^d \ mod \ N = M$.

7.2 b

Eve can ask Bob to sign $M^e * k^e \ mod \ N$, where Eve picks k coprime to N at random.By using Extended Euclid Even can find M. This way Bob's signature is distributed in the same way over all numbers inverted mod N.

8 Problem 2.4

8.1 a

Master Theorem:
$a = 5, b = 2, d = 1$
$a > b^d$
Run time:
$O(n^{log_b a}) = O(n^{log_2 5}) = O(n^{2.33})$

8.2 b

T(n) = 2T(n - 1) + C, for a constant C.
$C\Sigma_{i=0}^{n-1} \ 2^i + 2^n T(0) = O(2^n)$

8.3 c

Master Theorem:
$a = 9, b = 3, d = 2$
$a = b^d$
$O(n^d logn) = O(n^2 logn)$
I would choose this one.

9 Problem 2.12

Each time the function is called it prints out one line and calls the same function on half the size of the input. This means that the number of printed lines is:
$P(n) = 2P(\frac{n}{2}) + 1$

Using Master Theorem:
$P(n) = \Theta(n)$

10 Problem 2.23

10.1 a

In splitting the two arrays, we have A1 and A2. If A has a majority element, that element must also be a majority of A1 or A2 or both. To find the majority element, we recursively compute the majority elements of A1 and A2 and check whether any of these is a mahority element of A. The running time is $T(n) = 2T(\frac{n}{2}) + O(n) = O(nlogn)$

10.2 b

There are at most n/2 elements because at least one element in each pair is discarded. If the remaining elements have a majority element then there exist x of them within the elements appearing at least n/4 times. Therefore, x must have been paired up with itself in at least n/4 pairs, meaning thatqq there are at least n/2 of x. The running time for this is $T(n) = T(\frac{n}{2}) + O(n)$. Meaning $T(n) = O(n)$

11 Problem 3.2 (a)

$A \to B \to C \to D \to H \to G \to F \to E$

12 Problem 3.8

12.1 a

Let (directed) graph G = (V,E).
Triple nodes (a0, a1, a2)
Container sizes: C0 = 10, C1 = 7, C2 = 4
ai is the actual contents in the ith container.
$0 \leq ai \leq Si$ for i = 0, 1, 2 and at any given node a0 + a1 + a2 = 11. An edge between two nodes (a0, a1, a2) and (b0, b1, b2) exists if: (1) the two nodes differ in two coordinates and (2) if i, j are coordinates they differ in, then either ai = 0, or aj = 0 or ai = Si or aj = Sj.

12.2 b

Depth First Search Algorithm

12.3 c

On depth 6 on the depth first tree, the node (2, 7, 2) is reached which is the answer.