TJ Liggett

Izzy Sommers

COSC 330 Homework 2

Homework 0.4

1. Is the following Bare Bones program self-terminating? Explain your answer.

```
incr X        // X++
decr Y        // Y--
```

This program is self-terminating. *A Bare Bones program will self-terminate without any while loops. This program will increment X by one and decrement Y by one, assuming X and Y have been initialized, and then terminate.*

2. Is the following Bare Bones program self-terminating? Explain your answer.

```
Y=X
incr Y
incr Y              // Y = X + 2
while X not 0:
    decr X
    decr X
    decr Y

    decr Y
decr Y              // Y = At end of loop, Y will be 2 if X was even, 1 if Y was odd
while Y not 0:      // If X was even, Y = 1 and loop will run forever. Otherwise program terminates
```

If x is odd, this program will terminate. If x is even, this program will not terminate. *Y = X + 2 at the beginning of the first while loop. The first loop will terminate when X = 0, and Y will be decremented each time X is. However, if X is odd, there will be an extra decrement of Y, leaving Y to be 1 in this case. In this case, Y will be decremented again and Y = 0 for the second while loop, allowing the program to terminate.*

In the case X is even, Y will be equal to 2 at the end of the first loop, and thus equal to 1 at the beginning of the second loop, and the program will not terminate.

Homework 0.5

1. Suppose a problem can be solved by an algorithm in $O(n^2)$ as well as another algorithm in $O(2^n)$.

   Will one algorithm always outperform the other?

   One algorithm will not always outperform the other. With small values of n, an algorithm in O(2^n)

   can outperform an algorithm in $O(n^2)$. However, for some large n, eventually the in $O(n^2)$

   algorithm will outperform the O(2^n) algorithm for all values greater than n.

2. Give an example of a polynomial problem. Give an example of a nonpolynomial problem. Give an example of an NP problem that has yet has not been shown to be a polynomial problem.

   An example of a polynomial problem is sorting an integer array. For example, if we use an integer array using quick sort the time complexity is O(n log n) (radix sort could perform linearly but is worse in practice).

   An example of an NP problem is the Knapsack Problem: Given a set of items, each with weight and value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible. The decision problem here is NP-complete. That is, can a value of at least V be achieved without exceeding the weight W.

3. If the time complexity of algorithm X is greater than that of algorithm Y, is algorithm X necessarily harder to understand than algorithm Y? Explain your answer.

Not necessarily. The understandability of a program is its intellectual complexity. Time complexity and intellectual complexity are related, but not the same. Understandability by humans is viewed differently than complexity from a machine's point of view. It is possible that a program with a greater time complexity than another has a lower time complexity. For example, binary search and quick search can both be understood at the same complexity for a human, but their time complexity is not the same.

Homework 0.6

1. Find the factors of 66,043.

66043=1*____*____*____*…

66043 = 1 * 66043

66043 = 211 * 313

2. Using the public keys n = 91 and e = 5, encrypt the binary bit string 101. Give your answer in the form of a binary bit string.

Encrypt the binary bit string 101

101 binary → 5 decimal
5^5%91 = 3125% 91 = 31
31 decimal → 11111 binary

3. Using the private keys n = 91 and d = 29, decrypt the binary bit string 10. Give your answer in the form of a binary bit string.

Decrypt the binary bit string 10

10 binary ⇒ 2 decimal

2^{29}% 91 = 32

32 decimal ⇒ 100000

4. Find the appropriate value for the decrypting keys n and d in an RSA public-key cryptography system based on the primes p = 7 and q = 19 and the encryption key e = 5.

in = pq = 133

ed = k(p-1)(q-1) + 1 for some k

5d = k(6)(18) + 1

$d = \frac{108k+1}{5}$

d = 65 when k = 3

5. Write a program to read in an integer (long) and display all pairs of factors of the input integer. That is, the output must be given as a list of pairs of factors such that the product of each pair is the input integer. Then, print which of these pairs contains only prime numbers (there can be either 1 such pair or none). Note that
    1. the Scanner class supports a nextLong method to read a long integer, nextBigInteger to read a BigInteger, and nextBigDecimal to read a BigDecimal.
    2. the printf method uses %d for int, long, BigInteger; and %f for float, double, and BigDecimal.