HOMEWORK 7: §6.3 AND §7.1-7.4 DUE MARCH 9

Name: ______________________________

- Please refer to the syllabus regarding allowed collaboration on this homework assignment.
- All answers should be fully justified.
- Your homework should be neatly written on additional paper; you may attach this cover page if you would like to keep the questions attached to the answers.

(1) Suppose you want to have a program that evaluates polynomials.

(a) Here's (presumably) the simplest possible program.

```
Input:  polynomial p(x) = a_0 + a_1 x + a_2 x^2 + ··· + a_n x^n
        n, the degree
        α, a real number
Output: p(α) (the value of p at x = α)

Return( a_0 + a_1 · α + a_2 · α^2 + ··· + a_n · α^n )
```

How many additions and multiplications occur when this code is run? What is the asymptotic time complexity? (Important: exponentiation does not count as an "atomic operation"; you need to count the number of multiplications needed.)

(b) The above is pretty wasteful; why should the computer calculate α^{12} and later α^{13} from scratch? Write pseudocode that saves compuation time by looping through the terms of the polynomial, calculating successive powers of α more efficiently. What is its asymptotic time complexity?

Remark: there is an even more efficient algorithm, "Horner's Rule." It is slightly faster than the suggested algorithm in (1b).

(2) Suppose you have two algorithms, `blarg` and `wibble`, with time complexity $\Theta(n \log n)$ and $\Theta(n)$ respectively. `blarg` modifies the input, while `wibble` just checks something about the input and returns `True` or `False`. You write a new algorithm:

```
For i = 1 to n
    If wibble
        blarg
    End-if
End-for
```

Assume all calls to `blarg` and `wibble` occur on an input of size n.

(a) Suppose `wibble` always returns `True`. (It still takes $\Theta(n)$ time.) What is the time complexity of the algorithm?

(b) Suppose instead that `wibble` always returns `False`. What is the time complexity of the algorithm?

(c) Suppose instead that for the worst case input, `wibble` returns `True` for about $\log_5(n)$ values of i in the `For` loop. What is the time complexity of the algorithm?

(3) An important fact: if p is prime and $p \mid (ab)$, then $p \mid a$ or $p \mid b$. Is this true if p is not prime?

(4) Prove that if p is prime, then $\mathbb{Z}_p$ has the zero product property.

(5) Prove that if n is composite, then $\mathbb{Z}_n$ does not have the zero product property.

(6) (a) If $x \bmod 28 = 9$, then what is $6x \bmod 28$?

(b) If $y \bmod 15 = 4$, then what is $10y \bmod 3$?

A “joke”:

Theorem. *There are infinitely many primes.*

Proof idea. If not, take the product of all the primes and add 1. □

Theorem. *There are infinitely many composites.*

Proof idea. If not, take the product of all the composites and do not add 1. □