COMP8760

Class Worksheet 1

Sanjay Bhattacherjee

This worksheet primarily contains programming tasks. You may use any programming language of your choice.

1. **n-bit numbers**: Given n bits, they can represent 2^n different binary numbers. Following are examples.

n	Set of n-bit binary numbers
1	$\{0, 1\}$
2	$\{00, 01, 10, 11\}$
3	$\{000, 001, 010, 011, 100, 101, 110, 111\}$

   Write a program that takes as input an integer $n \leq 25$ and lists all possible binary values that may be represented using n bits.

2. We define the *floor* notation $\lfloor x \rfloor$ as the *largest integer smaller than or equal to* x. For example,

$$\begin{aligned}
\lfloor 20 \rfloor &= 20 \\
\left\lfloor \tfrac{20}{7} \right\rfloor &= 2 \\
\left\lfloor \tfrac{22}{7} \right\rfloor &= 3 \\
\left\lfloor \tfrac{22}{27} \right\rfloor &= 0 \\
\left\lfloor \tfrac{-22}{7} \right\rfloor &= -4 \\
\left\lfloor \tfrac{-20}{7} \right\rfloor &= -3 \\
\lfloor -20 \rfloor &= -20
\end{aligned}$$

   Write a program that will take as input a floating point number x and outputs $\lfloor x \rfloor$.

3. **Euclidean Algorithm**: Write a program that takes as input two positive integers a and b, and finds the greatest common divisor (gcd) of a and b using the Euclidean algorithm.
   Hint: The algorithm is described in Table 1.

Input:	a, b, assuming $a \geq b$
Output:	$\gcd(a, b)$
Initialisation:	$r \leftarrow b$
Iteration:	`while` $r \neq 0$ `do:`
	$r \leftarrow a\%b$ // % is the operator for finding the remainder
	$a \leftarrow b$
	$b \leftarrow r$
Final:	`return` a

Table 1: The Euclidean algorithm for finding $\gcd(a, b)$.

4. **Exponentiation**: The following tasks are on computing exponentiations with progressively better efficiencies.

(a) **Naive exponentiation.** Write a program that takes as input an integer n and a positive integer e and computes n^e using $e-1$ multiplications.

$$n^e = \underbrace{n \times n \times \cdots \times n}_{e \text{ times}}$$

(b) **Recursive exponentiation.** Write a program that takes as input an integer n and a positive integer e and computes n^e recursively as follows.

$$n^e \leftarrow \begin{cases} 1 & \text{if } e = 0, \\ n & \text{if } e = 1, \\ n^{\frac{e}{2}} \times n^{\frac{e}{2}} & \text{if } e \text{ is even, and} \\ n \times n^{\frac{e-1}{2}} \times n^{\frac{e-1}{2}} & \text{if } e \text{ is odd.} \end{cases}$$

What is the maximum number of multiplications that would be required for computing n^e?

(c) **Non-recursive smart exponentiation.** Write a program that takes as input an integer n and a positive integer e and computes n^e non-recursively using at most $\log_2 e$ multiplications.
Hint: Find the binary representation of the integer e and use that in the computation of n^e.

5. **Groups**: We recollect here that for a positive integer N, the set of all remainders of N is denoted as

$$\mathbb{Z}_N = \{0, 1, 2, \ldots, N-1\}.$$

This set forms a group under the operation *addition modulo* N. The set $\mathbb{Z}_N^\star$ contains those integers between 1 and $N-1$ that are mutually prime to N. More formally,

$$\mathbb{Z}_N^\star = \{x : \gcd(x, n) = 1\}.$$

This set forms a group under the operation *multiplication modulo* N.

Write a program that will take as input a positive integer N and will give the following outputs:

(a) The table denoting all addition operations between the elements of the group $\mathbb{Z}_N$.

(b) For every $a \in \mathbb{Z}_N$, the values of ka, for all $1 \leq k \leq |\mathbb{Z}_N|$.
Note that $\mathbb{Z}_N$ is an additive group. So,

$$ka = \underbrace{a + a + \cdots + a}_{k \text{ times}} \mod N.$$

(c) The elements of the group $\mathbb{Z}_N^\star$.

(d) For every $a \in \mathbb{Z}_N^\star$, the values of a^i, for all $1 \leq i \leq |\mathbb{Z}_N^\star|$.
Note that $\mathbb{Z}_N^\star$ is a multiplicative group. So,

$$a^i = \underbrace{a \times a \times \cdots \times a}_{i \text{ times}} \mod N.$$

(e) The table denoting all multiplication operations between the elements of the group $\mathbb{Z}_N^\star$.

Experiments: Try executing the program you have written with values of N such that some of those are prime numbers (say $N = 5, 7, 11, 13$) and the others are composite numbers (say $N = 4, 6, 8, 9, 10, 12$). Observe and note the differences between these two cases.