**CHAPTER 0-1**

**functions and their computation**

* A computer program computes a function.
  + A function is a mapping between a set of possible input values and a set of output values so that each possible input is mapped to a single output.
* The process of determining the particular output value that a function assigns to a given input is called computing the function.
* A fundamental task of computer science is to find techniques for computing the functions that lie beneath the problems we want to solve.

**Computing a function using a table**

* One simple technique for computing a function is to represent the function’s inputs and outputs in a table.
  + Each time the output of the function is required, we merely look for the given input in the table where we find the required output.
* As an example of a function that can be represented completely in tabular form, consider the function that converts a decimal digit into a binary bit string.
  + We can simply construct the following table:

Decimal digit (input) binary bit string (output)

0 0

1 1

2 10

3 11

4 100

5 101

6 110

7 111

8 1000

9 1001

* The table technique is convenient but limited in power because many functions cannot be represented completely in tabular form.
* As an example of a function that cannot be represented completely in tabular form, consider the function that converts a decimal number into a binary bit string.
  + Because there is no limit to the list of possible input/output pairs, the table is destined to be incomplete.

**Computing a function using an algebraic formula**

* A more powerful technique to computing functions is to follow directions provided by an algebraic formula.
* As an example of a function that can be computed by an algebraic formula, consider the function to compute the value of an investment of P after earning an annually compounded interest rate of r for n years. This function can be computed by the following formula:

V=P(1+r)n

* But the expressive power of algebraic formulas has its limitations as well. There are functions whose input/output relationships are too complex to be described by algebraic manipulations.
* As an example of a function that cannot be computed by an algebraic formula, consider the function that converts a decimal number into a binary bit string.

**Computing a function using an algorithm**

* As we consider functions whose input/output relationships are more and more complex, we are forced to represent the computation in the form of an algorithm.
* As an example, we can certainly write a step-by-step procedure to convert a decimal number into a binary bit string.

**Noncomputable functions**

* A striking result from mathematics is that there are functions that are so complex that there is no well-defined, step-by-step process for determining their outputs based on their input values.
  + Thus, the computation of these functions lies beyond the abilities of any algorithmic system.
  + These functions are said to be noncomputable, whereas the functions whose output values can be determined algorithmically from their input values are said to be computable.
* The distinction between computable and noncomputable functions is important in computer science.
  + Because computers can only perform tasks described by algorithms, the study of computable functions is the study of the ultimate capabilities of computers.
  + If we discover that the solution to a problem requires the computation of a noncomputable function, we can conclude that the solution to that problem lies beyond the capabilities of computers.

**HW**

1. Identify one function that can be represented completely in tabular form. Show the table.
2. Identify one function whose outputs can be described as an algebraic expression involving their inputs. Show the algebraic expression.
3. Identify a function that cannot be described in terms of an algebraic formula.