3. Algorithms

Department of Mathematics & Statistics

ASU

Outline of Chapter 3

1. **Algorithms**
2. **The Growth of functions**
3. **Complexity of Algorithms**

- In this chapter, we will introduce algorithms for searching for an element in a list and sorting a list so its elements are in some prescribed order.
- One of the most important issues on algorithms is their computational complexity, i.e., we need to consider the computational time and computer memory, whenever we solve scientific problems.

- A general meaning of an algorithm is to include all definite procedures for solving problems.
- Instead of using a particular computer language such as C, $C++$, FORTRAN, or Java to specify algorithms, it will be better to use a Pseudocode which is a common way to describe the algorithms.
- A **Pseudocode** provides an intermediate step between an English language description of the steps of a procedure and a specification of the procedure. The advantage of using pseudocode is to make programers to understand algorithms easily, because of its simplicity and common use.

- Here is a good example of a pseudocode describing the algorithm for finding the maximum in a finite sequence.

Algorithm1

Finding the Maximum element in a Finite Sequence in $\mathbb{Z}$.

procedure $\max(a_1, a_2, \cdots, a_n)$

$\max := a_1$

for $i := 2$ **to** n

  if $\max < a_i$ **then** $\max := a_i$

return max

- **Properties of Algorithms** are useful to keep in mind.

1. **Inputs:** All data that we need, in order to solve a problem.
2. **Outputs:** are the solution to the problem.
3. **Definiteness:** the steps of an algorithm must be defined.
4. **Correctness:** an algorithm should produce the correct output.
5. **Finiteness:** an algorithm should produce the desired output after finite steps.
6. **Effectiveness:** an algorithm should perform each step exactly.

Linear (Sequential) Search Algorithm

- The problems of locating an element in an ordered list are called **searching problems.**

Algorithm2

The Linear Search Algorithm

procedure linear search $(x : \textbf{integer}, a_1, a_2, \cdots, a_n)$ with $a_i \neq a_j$

$i := 1$

while $(i \leq n$ and $x \neq a_i)$

$\quad i := i + 1$

if $i \leq n$ **then** $location := i$

else $location := 0$

return $location\{location$ is the subscript of the term that equals x, or is 0 if x is not found$\}$

Algorithm3

The Binary Search Algorithm

procedure binary search $(x : \mathbf{integer}, a_1, a_2, \cdots, a_n)$ with increasing integers

$i := 1\{i$ is left endpoint of search interval$\}$

$j := n\{j$ is right endpoint of search interval$\}$

while $i < j$

$\quad m := \lfloor (i + j)/2 \rfloor$

$\quad$ **if** $x > a_m$ **then** $i := m + 1$

$\quad$ **else** $j := m$

if $x = a_i$ **then** $location := i$

else $location := 0$

return $location\{location$ is the subscript of the term that equals x, or is 0 if x is not found$\}$

- Solving optimization problems means finding a minimum value or maximum value. Algorithms that make a best choice at each step are called **greedy algorithms.**

Algorithm4

Greedy Change-Making Algorithm
procedure change $(c_1, c_2, \cdots, c_r)$**:** values of denomination of coins, where $c_1 > c_2 > \cdots > c_r$; n : a positive integer.
for $i := 1$ **to** r
    $d_i := 0\{d_i$ counts the coins of denomination c_i used$\}$
    while $n \geq c_i$
        $d_i := d_i + 1\{$add a coin of denomination $c_i\}$
        $n := n - c_i$
$\{d_i$ is the number of coins of denomination c_i in the change for $i = 1, 2, \cdots, r\}$

Example1

1. List all the steps used to search for 9 in the sequence $\{1, 3, 4, 5, 6, 8, 9, 11\}$ using
(1) a linear search (2) a binary search
2. Devise an algorithm that finds the sum of all the integers in a list.
3. Describe an algorithm for finding the smallest integer in a finite sequence in $\mathbb{N}$.

Example2

Use the greedy algorithm to make change using quarters, dimes, nickles, and pennies for
(1) 50 cents (2) 65 cents (3) 77 cents (4) 80 cents.