Data Structures I

Week # 2: Algorithm

Presented

by

Dr. Nadia M. Mohammed

What is algorithm?

- A finite set of instructions which accomplish a particular task
- A method or process to solve a problem
- Transforms input of a problem to output

Algorithm = Input + Process + Output

Algorithm development is an art – it needs practice, practice and only practice!

What is a good algorithm?

- It must be correct
- It must be finite (in terms of time and size)
- It must terminate
- It must be unambiguous
  - Which step is next?
- It must be space and time efficient

A program is an instance of an algorithm, written in some specific programming language

A simple algorithm

- Problem: Find maximum of *a*, *b*, *c*
- Algorithm
  - Input = *a*, *b*, *c*
  - Output = *max*
  - Process

    Let *max* = *a*

    If *b* > *max* then

    max = *b*

    If *c* > *max* then

    max = *c*

    Display *max*

Order is very important!!!

Algorithm development: Basics

- Clearly identify:
  - What output is required?
  - What is the input?
  - What steps are required to transform input into output
    - The most crucial bit
    - Needs problem solving skills
    - A problem can be solved in many different ways
    - Which solution, amongst the different possible solutions is optimal?

Even/ Odd Numbers

```
Input  range
for num←–0; num<= range; num←–num+1 do
   if num % 2 = 0 then
        print num is even
  else
        print num is odd
  endif
endfor
```

Computing weekly wages

```
Input hours_worked, pay_rate
if hours_worked <= 40 then
   gross_pay ← pay_rate * hours_worked
else
  basic_pay ← pay_rate * 40
  over_time ← hours_worked – 40
  over_time_pay ← 1.5 * pay_rate * over_time
  gross_pay ← basic_pay + over_time_pay
endif
print gross_pay
```

Homework

1. Write an **algorithm** to **find the largest** of a set of numbers. You do not know the number of numbers.

2. Write an **algorithm** that **finds the average** of (n) numbers.

   For example numbers are [4,5,14,20,3,6]