Chapter 2
Algorithm Discovery and Design

Wen-Hung Liao, 10/5/2001
Representing Algorithms

Possible choices

- Natural language (Figure 2.1)
- Formal programming language (Figure 2.2)

Our choice: pseudocode

- A special set of English language constructs modeled to look like the statements available in most programming languages.
- A compromise between natural language and formal programming language
Sequential Operations

Three basic sequential operations: computation, input and output.

Computation:
Set the value of "variable" to "arithmetic expression"

Input: Get values for "variable", "variable", ...

Output: Print the values of "variable", "variable", ...
Example: Computing mpg

- Step 1: Get values for gallons used, starting mileage, ending mileage
- Step 2: Set the value of distance driven to (ending mileage - starting mileage)
- Step 3: Set the value of average miles per gallon to (distance driven / gallons used)
- Step 4: Print the value of average miles per gallon
- Step 5: Stop
Conditional Operations

- If/then/else:
  - if “a true/false condition” is true then
    - first set of algorithmic operations
  - else (otherwise)
    - second set of algorithmic operations
Iterative Operations

Looping: repetition of a block of instructions
Repeat step i to step j until “a true/false” condition becomes true
step i:  operation
step i+1: operation
...
step j:  operation
Terminology

- Loop body
- Termination condition
- Infinite loop
Example of a loop

- **Step 1:** Set the value of `count` to 1
- **Step 2:** Repeat step 3 to step 5 until `count > 100`
  - **Step 3:** Set `square` to `(count x count)`
  - **Step 4:** Print the values of `count` and `square`
  - **Step 5:** Add 1 to `count`
Variations of a loop

- Repeat/until
- While "a true/false condition" remains true do

Figure 2.6: summary of pseudocode language instructions
Examples of Algorithmic Problem Solving

- Process of finding a solution to a given problem is called algorithm discovery.
- Sequential search: searching an unordered list of values.
- First attempt: Figure 2.7
- Second attempt: Figure 2.8: using iteration.
- Third attempt: Figure 2.9, consider all possibilities.
Search an ordered list of values

- Dictionary, Telephone book
- Data are organized in a certain manner.
- A different search technique should be used.

The selection of an algorithm to solve a problem is greatly influenced by the way the data are organized.
Example 2: Find largest

Problem statement:
Given a value \( n \geq 2 \) and a list containing exactly \( n \) unique numbers called \( A_1, A_2, \ldots, A_n \), find and print out both the largest value in the list and the position in the list where that largest value occurred.

“Find largest” can be a building block for the construction of solutions to other problems, such as sorting.
Find Largest Algorithm

- Get a values for n, the size of the list
- Get values for A_1, A_2, ..., A_n
- Set the value of largest so far to A_1
- Set the value of location to 1
- Set the value of i to 2
- Repeat until i > n
  - If A_i > largest so far then
    - Set largest so far to A_i
    - Set location to i
  - Add 1 to the value of i
- End of loop
- Print out the values of largest so far and location
- Stop
Example 3: Pattern Matching

- **Text search, genome analysis**
- **Problem statement**: You will be given some text composed of $n$ characters that will be referred to as $T_1, T_2, \ldots, T_n$. You will also be given a pattern of $m$ characters, $m \leq n$, that will be represented as $P_1, P_2, \ldots, P_m$. The algorithm must locate every occurrence of the pattern within the text. The output of the algorithm is the location of the text where each match occurred. For this problem, the location of the match is defined to be the index position in the text where the match begins.
Example

- Text: to be or not to be, that is the question
- Pattern: to
- Output: match at positions 1, 14.
- Step 1: the matching process
- Step 2: the slide forward
Pattern Matching: 1st draft

- Figure 2.11
- Attempt to match every character in the pattern beginning at position \( k \)
- Does the above statement violates the requirement discussed in Chapter 1?
- OK for initial phase.
- Use abstraction, top-down design.
Pattern Matching: final draft

- Figure 2.12
- Fill in the details
Homework #2

Problem 1, page 63.

Use an example to verify that your algorithm works correctly, say, take $f(x) = x^3 - x^2 + 2x - 2$, starting point $x=0$, and step size, accuracy of your choice.

Due on 10/12/2001.