Ch. 2 Algorithmic Foundations

Designing algorithms:

I. Express solution at an abstract level step-by-step in plain English

1. \( a \times b \) Add \( a \) \( b \) times or

   \[ \text{Algorithm 1} \]

   \[ 357 \]

   \[ 2 \]

   \[ 714 \]

   \[ \text{Algorithm 2} \]

   \[ \text{Algorithm 2 uses fewer steps; scales better} \]

   \[ \text{Number of steps grows logarithmically with number of digits in Algorithm 2, whereas number of steps can be very large for Algorithm 1.} \]

   \[ \text{Design \rightarrow analysis can be done without coding} \]

   \[ \text{Time taken by algorithm can be estimated by total number of operations} \]

   \[ \text{Design \rightarrow analyze \rightarrow build} \]

   \[ \text{Operation types:} \]

   - Sequential: series of ordered steps, no branching

   - TBD: Expand steps to more detailed sub-steps as needed. Write pseudocode for better precision.

   - Finding gas mileage:

     1. distance traveled
     2. gas used
     3. mileage \leftarrow \text{distance/gallons used}
Flowchart for sequential operation - finding gas mileage

Input: initial & final readings

Distance: Final - initial readings

Input: gallons used

Output: mileage

"Straight line algorithm"

Conditional:
If < condition > then ... else ...

Flowchart for conditional

If mileage > 40 mpg then output "Great Mileage!!!"
else /* no output */
Iterative:

While < condition >, step 1, step 2, ... and while

1. Input: number of records
   (Record) (Total)

2. Initialize (Average) at 0
   (Count) at 0

3. While (Count < Record)

   3.1: Input (Initial Reading),
        (Final Reading), (Gas Used)

   3.2: Distance ← Final Reading - Initial Reading

3.3: Mileage ← Distance / Gas - Used

3.4: Average ← Average + Mileage

3.5: Count ← Count + 1
   end while

4: Calculate average by dividing accumulated
   Total by count.

   Average ← Total / count

This pseudocode will fail for 0 records. Where?

Can insert after Step 1:

If (Record > 0) then
   Carry out Steps 2 through Step 4
else
   Output "Number of records must be at least 1."