Algorithm Discovery & Design

Garrett Dancik, PhD

Fall 2024

Course Notes: https://gdancik.github.io

Computer Science & Algorithms

- An algorithm is a step-by-step set of instructions for solving a task
- Computer Science is the study of algorithms, including
  - their formal and mathematical properties

    Is it correct? How long does it take?
  - their hardware realizations

    How can we design and build a computer to carry out it?
  - their linguistic realizations

    How can we design and use programming languages to implement them?
  - their applications

    What software can be developed? Operating Systems? Databases? AI? The Internet?

Example Algorithm (not great)

- Algorithm for Shampooing your Hair
  - Step 1: Wet your hair
  - Step 2: Lather
  - Step 4: Repeat

But this is is not a good algorithm:

- Step 4 is ambiguous (it is not clear what statements should be repeated, nor how many times)
- Step 1 may be clarification (how “wet” does our hair need to be)
- Step 2 may need clarification (how do we “lather”)

Algorithm: more precise definition

- An algorithm is a well-ordered collection of unambiguous and effectively computable operations that, when executed, produces a result and halts in a finite amount of time.

A correct solution to the shampooing problem

Step	Operation
1	Wet your hair
2	Set the value of *WashCount* to 0
3	Repeat Steps 4 through 6 until the value of *WashCount* equals 2
4	Lather your hair
5	Rinse your hair
6	Add 1 to the value of *WashCount*
7	Stop, you have finished shampooing your hair

Note: If any step is not “computable”, more information could be provided:

Step 4: Lather your hair

- 4.1. Apply 2 teaspoons of shampoo to the palm of your hand
- 4.2. Rub your hands together to create a lather (with visible bubbles)
- 4.3 Use the tips of your fingers to massage the shampoo into your scalp, for 1 – 3 minutes

Another correct solution to the shampooing problem

Step	Operation
1	Wet your hair
2	Lather your hair
3	Rinse your hair
4	Lather your hair
5	Rinse your hair
6	Stop, you have finished shampooing your hair

There will *always* be more than one algorithm for a problem, but some may be

- More readable
- Easier to change
- More efficient (are quicker to complete)

Pseudocode

- Pseudocode consists of English language constructs that resemble computer code, but that does not run on a computer. It is a compromise between natural language (which can be wordy and imprecise) and computer code (which may be too detailed).
- We start with three *sequential operations* (each operation performs a single, well-defined task)
  - Computation: set a variable (a named storage location) to the value of an expression
    - total ← x + y. (set total to the sum of *x* and *y; x* and *y* would need to have values)
  - Input: get data values from outside the program (from a file, from a sensor, from the keyboard)
    - Get values for “x” and “y”
  - Output: Send results to the outside world (the screen or display)
    - Print the values of “x” and “y”

Representing Algorithms Sequential Algorithm

FIGURE 2.3

Step	Operation
1	Get values for *gallons used, starting mileage, ending mileage*
2	Set value of *distance driven* to *(ending mileage - starting mileage)*
3	Set value of *average miles per gallon* to *(distance driven ÷ gallons used)*
	Print the value of *average miles per gallon*
5	Stop

Algorithm for computing average miles per gallon

Representing Algorithms Sequential Algorithm

FIGURE 2.3

Step	Operation
1	Get values for *gallons used, starting mileage, ending mileage*
2	Set value of *distance driven* to *(ending mileage - starting mileage)*
3	Set value of *average miles per gallon* to *(distance driven ÷ gallons used)*
	Print the value of *average miles per gallon*
5	Stop

Step	Operation
1 2 3	*gallons used* ← get value from user *starting mileage* ← get value from user *ending mileage* ← get value from user
4	*distance* ← *ending mileage – starting mileage*
5	*average miles per gallon* ← *distance driven / gallons used*
6	Print *average miles per gallon*
7	Stop

Algorithm for computing average miles per gallon

(notation on right is more common)

Example

- What is an algorithm for converting a number in *meters* to a number in *yards*? (Note: in the 2024 Olympics, USA sprinter set a world record of 9.79 seconds for the 100m). There are 1.093613 yards in a meter.

Conditional Statements

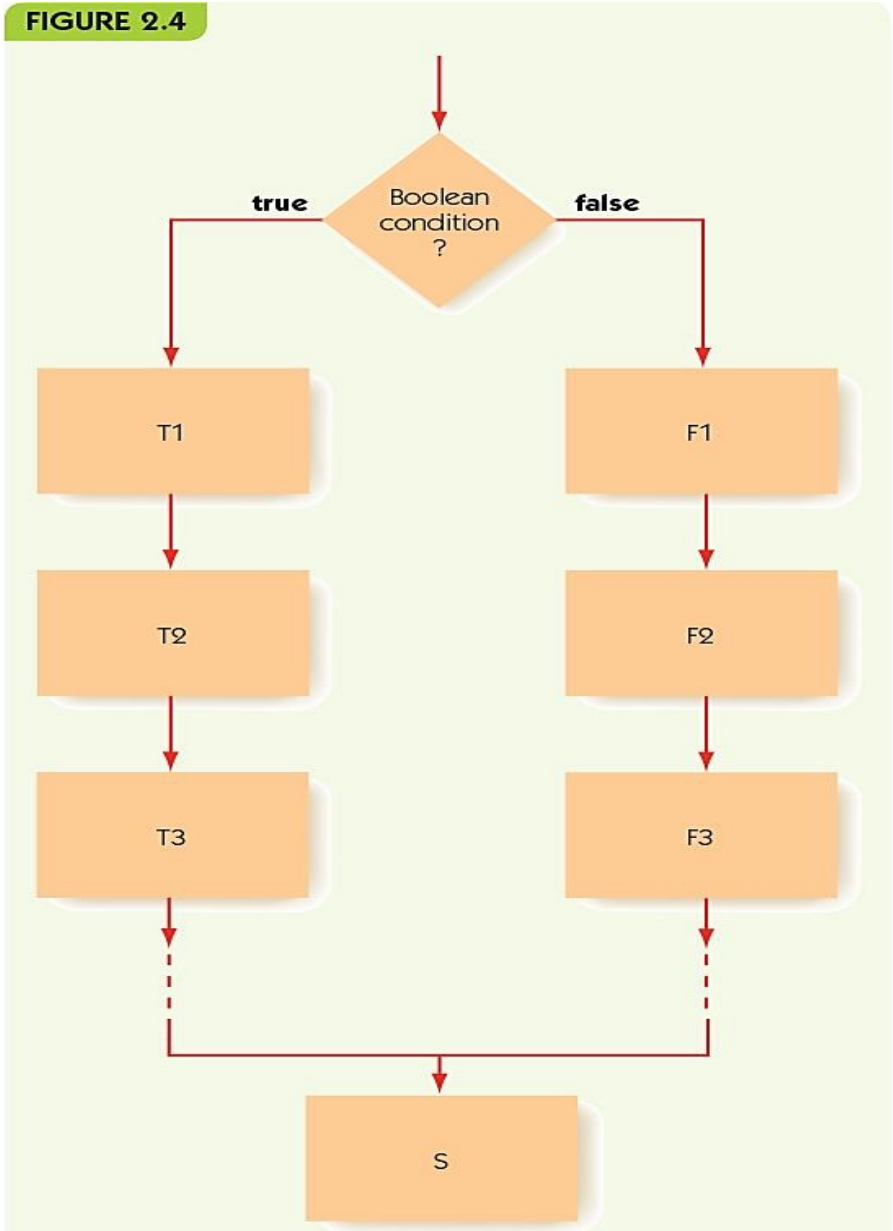
Control operations

- changes the normal flow of control, which is linear.
- Include conditional and iterative statements

Conditional statement: asks a question and selects among alternative options:

1. Evaluate the true/false condition
2. If the condition is true, then do the first set of operations and skip the second set
3. If the condition is false, skip the first set of operations and do the second set

Example: check for good or bad gas mileage



An Algorithm with a Conditional Statement

Problem: check for good or bad gas mileage. We consider 25 or more *mpg* to be good.

FIGURE 2.5

Step	Operation
1	Get values for *gallons used, starting mileage, ending mileage*
2	Set value of *distance driven* to (*ending mileage – starting mileage*)
3	Set value of *average miles per gallon* to (*distance driven ÷ gallons used*)
4	Print the value of *average miles per gallon*
5	If *average miles per gallon* is >25.0 then
6	Print the message 'You are getting good gas mileage'
	Else
7	Print the message 'You are NOT getting good gas mileage'
8	Stop

Second version of the average miles per gallon algorithm

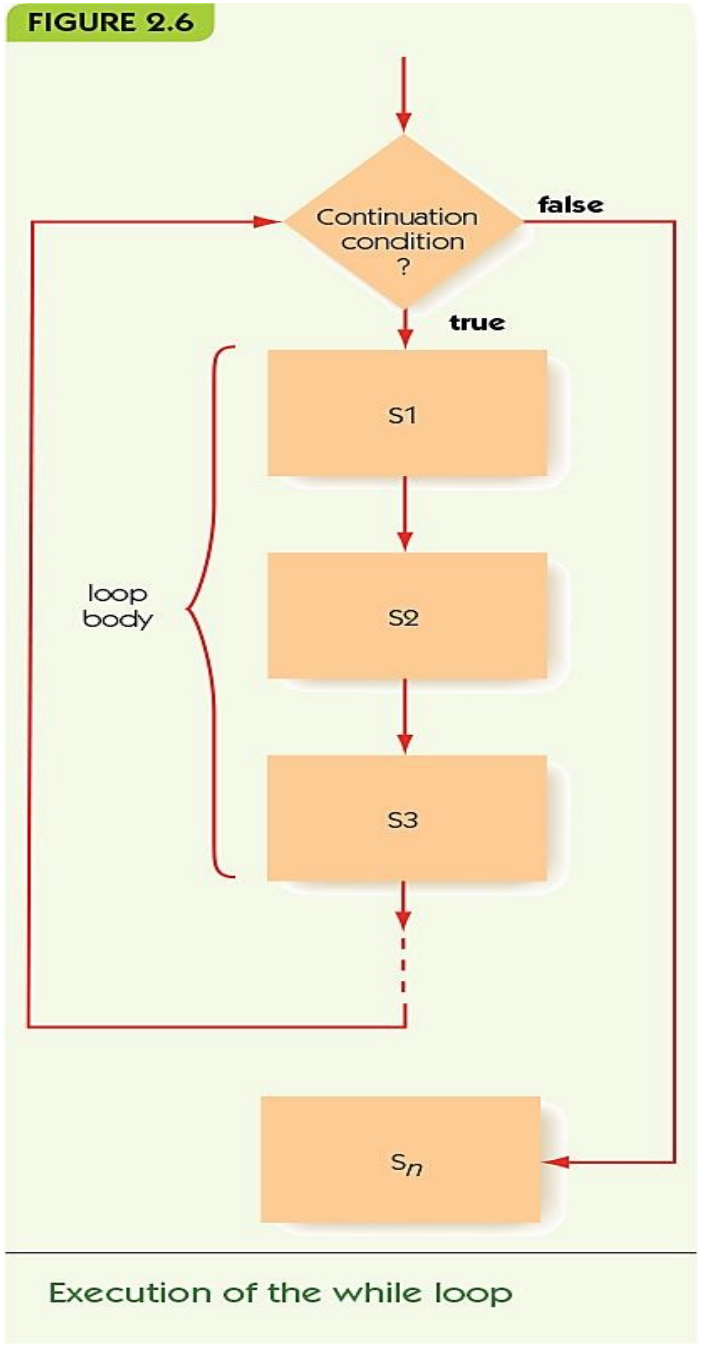
Iterations

An iteration is an operation that causes looping, which repeats a block of instructions

While statement repeats while a condition remains true

- **Continuation condition**: a test to see if while loop should continue
- **Loop body**: instructions to perform repeatedly

Example: repeated mileage calculations



An Algorithm with Iteration and Loop Body

Problem: check for good or bad gas mileage, *as many times as the user wants*. We consider 25 or more *mpg* to be good.

FIGURE 2.7

Step	Operation
1	*response* = Yes
2	While (*response* = Yes) do Steps 3 through 11
3	Get values for *gallons used*, *starting mileage*, *ending mileage*
4	Set value of *distance driven* to (*ending mileage* – *starting mileage*)
5	Set value of *average miles per gallon* to (*distance driven* ÷ *gallons used*)
6	Print the value of *average miles per gallon*
7	If average miles per gallon > 25.0 then
8	Print the message 'You are getting good gas mileage'
	Else
9	Print the message 'You are NOT getting good gas mileage'
10	Print the message 'Do you want to do this again? Enter Yes or No'
11	Get a new value for *response* from the user
12	Stop

Third version of the average miles per gallon algorithm

Examples of Algorithmic Problem Solving: Sequential Search

Problem: Determine the name associated with a given phone number.

Let $T_1 \ldots T_{10,000}$ represent 10,000 phone numbers in a database

Let $N_1 \ldots N_{10,000}$ represent the 10,000 names corresponding to the 10,000 phone numbers

FIGURE 2.13

Step	Operation
1	Get values for *NUMBER*, $T_1, \ldots, T_{10,000}$, and $N_1, \ldots, N_{10,000}$
2	Set the value of *i* to 1 and set the value of *Found* to NO
3	While both (*Found* = NO) and ($i \leq 10,000$) do Steps 4 through 7
4	If NUMBER is equal to the ith number on the list T_i then
5	Print the name of the corresponding person, N_i
6	Set the value of *Found* to YES
	Else (*NUMBER* is not equal to T_i)
7	Add 1 to the value of *i*
8	If (*Found* = NO) then
9	Print the message 'Sorry, this number is not in the directory'
10	Stop

The sequential search algorithm

Example

- Find the maximum of 2 numbers

- Find the maximum of 10,000 numbers