CMPS1134
Fundamentals of Computing

Algorithms 1

Computer Science: An Overview
Eleventh Edition
J. Glenn Brookshear
Chapter 5

Chapter 5: Algorithms

- **The Concept of an Algorithm**
- **Algorithm Representation**
- **Algorithm Discovery**
- **Iterative Structures**

- Recursive Structures
- Efficiency and Correctness

Definition of Algorithm

An algorithm is an **ordered** set of **unambiguous**, **executable** steps that defines a **terminating** process.

Algorithm Representation

- Requires well-defined primitives
- A collection of primitives constitutes a programming language.

Figure 5.2 **Folding a bird from a square piece of paper**

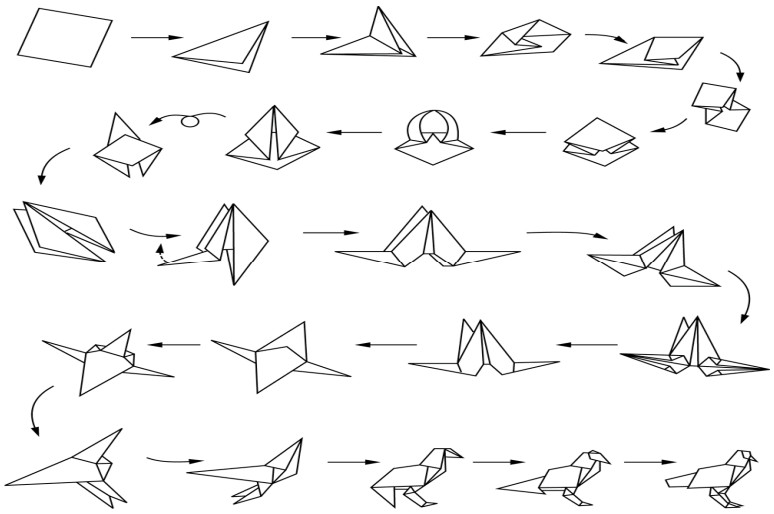
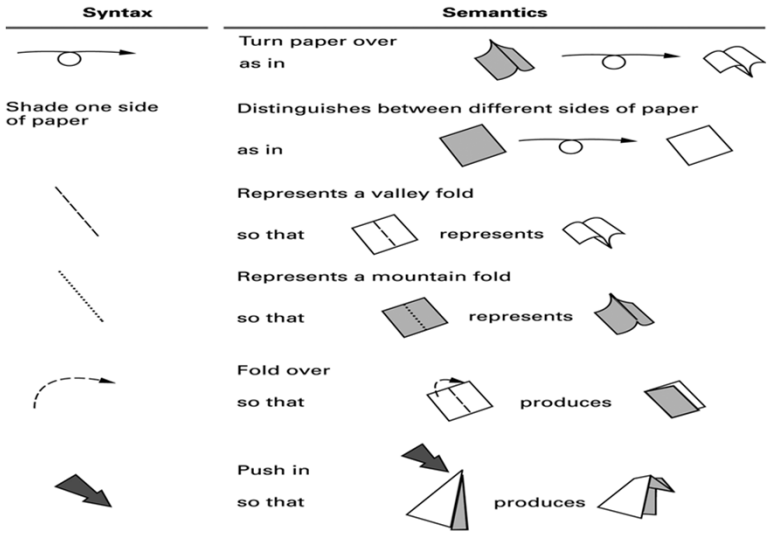


Figure 5.3 **Origami primitives**



Pseudocode Primitives

- Assignment

  name ← *expression*

  e.g.

  sum ← total1 + total 2

- Conditional selection

  if (condition) **then** (activity) **else** (activity)

  e.g.

  if (not raining)

  then (go for a walk)

  else (watch television)

Pseudocode Primitives (cont)

- Repeated execution

  while (*condition)* **do** (*activity)*

  e.g.

  while (tickets remain to be sold) **do**

  (sell a ticket)

- Procedure

  procedure *name* (*parameters*)

  e.g.

  procedure ProcessLoan

  procedure Sum (total1, total2)

Pseudocode Primitives (cont)

- Indentation shows **nested** conditions

```
if (not raining)
   then (if (temperature = hot)
            then (go swimming)
            else (play golf))
   else (watch television)
```


Figure 5.4 **The procedure Greetings in pseudocode**

```
procedure Greetings
Count ← 3;
while (Count > 0) do
    (print the message "Hello" and
        Count ← Count –1)
```


Polya's Problem Solving Steps

1. Understand the problem.
2. Devise a plan for solving the problem.
3. Carry out the plan.
4. Evaluate the solution for accuracy and its potential as a tool for solving other problems.

Getting a Foot in the Door

- Try working the problem backwards
- Solve an easier related problem
  - Relax some of the problem constraints
  - Solve pieces of the problem first (bottom up methodology)
- Stepwise refinement: Divide the problem into smaller problems (top-down methodology)

Ages of Children Problem

- Person A is charged with the task of determining the ages of B's three children.
  - B tells A that the product of the children's ages is 36.
  - A replies that another clue is required.
  - B tells A the sum of the children's ages (we are not told what it is)
  - A replies that another clue is needed.
  - B tells A that the oldest child plays the piano.
  - A tells B the ages of the three children.
- How old are the three children?

a. Triples whose product is 36

(1,1,36) (1,6,6)
(1,2,18) (2,2,9)
(1,3,12) (2,3,6)
(1,4,9) (3,3,4)

b. Sums of triples from (a)

1+1+36=38
1+2+18=21
1+3+12=16
1+4+9=14
1+6+6=13
2+2+9=13
2+3+6=11
3+3+4=10

Iterative Structures

- **Pretest loop**:

```
while (condition) do
   (loop body)
```

No assumption made for first iteration

```
while (there is a coin in your pocket) do
   (take coin from your pocket)
```

- **Posttest loop**:

```
repeat (loop body)
   until(condition)
```

Assumption made for first iteration

```
repeat (take a coin from your pocket)
   until(there are no coins in your pocket)
```


Figure 5.7 **Components of repetitive control**

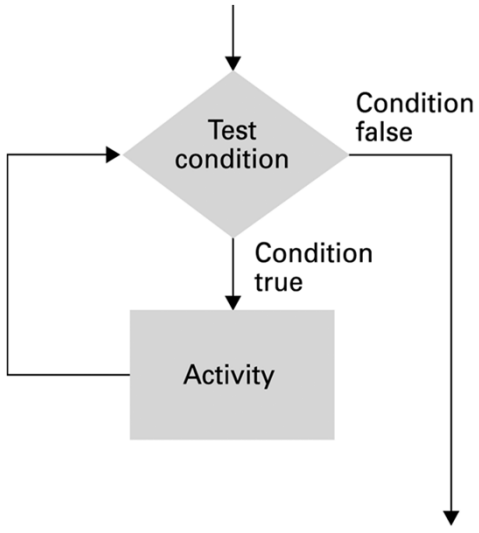
Initialize: Establish an initial state that will be modified toward the termination condition

Test: Compare the current state to the termination condition and terminate the repetition if equal

Modify: Change the state in such a way that it moves toward the termination condition

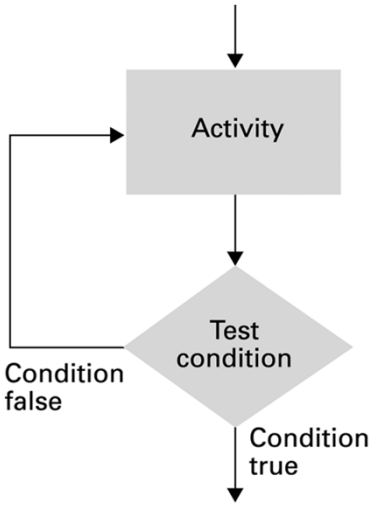
 15

Figure 5.8 **The while loop structure**



 16

Figure 5.9 **The repeat loop structure**



 17

Figure 5.6 **The sequential search algorithm in pseudocode**

```
procedure Search (List, TargetValue)
  if (List is empty)
    then (Declare search a failure)
    else
          (Select the first entry in List to be TestEntry)
          while (TargetValue > TestEntry and entries remain)
            do (Select the next entry in List as TestEntry)
          if (TargetValue = TestEntry)
            then (Declare search a success)
            else (Declare search a failure)
```

 18

Figure 5.10 **Sorting the list Fred, Alex, Diana, Byron, and Carol alphabetically**

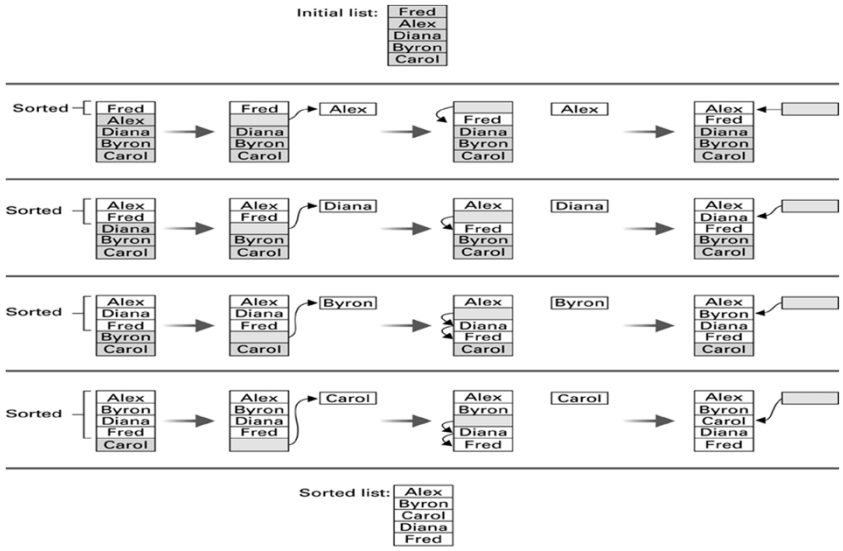


Figure 5.11 **The insertion sort algorithm expressed in pseudocode**

```
procedure Sort (List)
N ← 2;
while (the value of N does not exceed the length of List) do
    (Select the Nth entry in List as the pivot entry;
    Move the pivot entry to a temporary location leaving a hole in List;
     while (there is a name above the hole and that name is greater than the pivot) do
          (move the name above the hole down into the hole leaving a hole above the name)
    Move the pivot entry into the hole in List;
    N ← N + 1
    )
```

