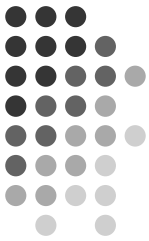
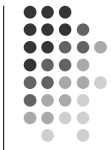
Algorithms: The recipe for computation

2C

Visualizing Algorithms with Flowcharts

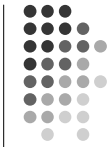


Flowcharts

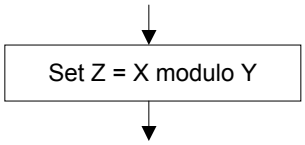


- Flowcharts are used to show the flow of control of an algorithm as it runs step by step in a visual manner.
- Flowcharts consist of the following components:
  - An oval labeled START
  - A sequence of boxes with algorithm operations
  - Arrows that indicate the order that the boxes are evaluated in the algorithm
  - A oval labeled STOP or END

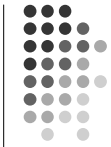
Sequential Flow



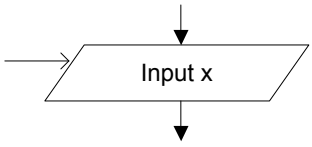
- An assignment operation sets a variable to a specific value or changes a variable to a new value.
  - Represented in a flowchart using a rectangle with the assignment operation described inside
  - Example:



Input

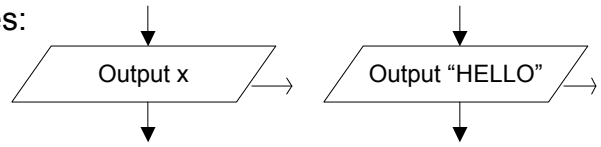


- An input operation sets a variable to a data value given by the user of the algorithm.
  - Represented in a flowchart using a rhombus with a small arrow pointing in from the side
  - The contents of the rhombus include the variable being initialized
  - Example:



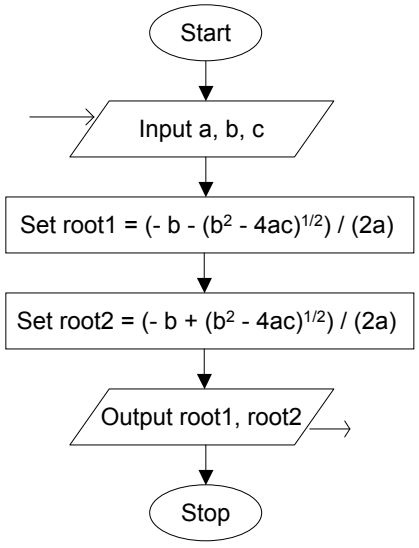
Output

- An output operation displays the data value in a variable or a message to the user of the algorithm.
  - Represented in a flowchart using a rhombus with a small arrow pointing out from the side
  - The contents of the rhombus include the variable or message being displayed
  - Examples:



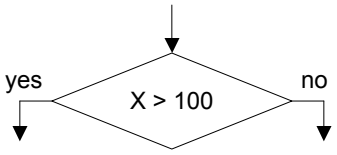
Example

- Algorithm to find the roots of a quadratic equation $ax^2 + bx + c = 0$



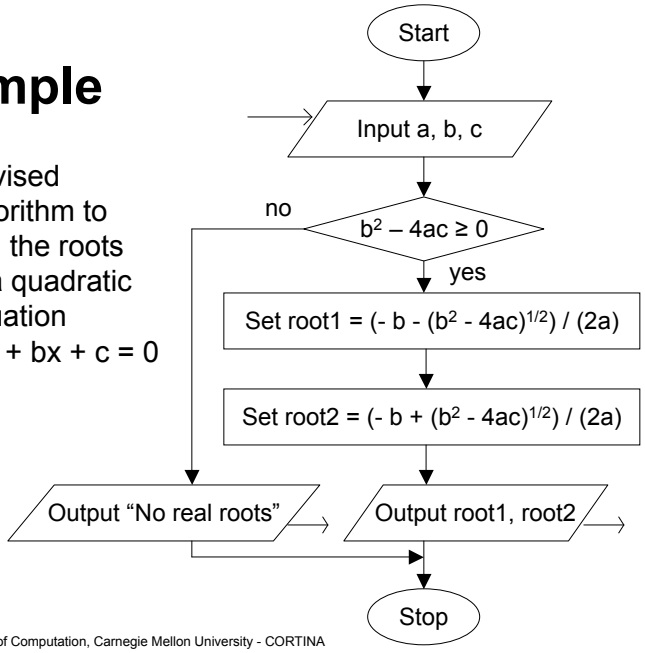
Conditional Operations

- A conditional operation determines the direction of flow in an algorithm
  - Represented in a flowchart using a diamond with a test condition inside
  - Example:



Example

- Revised algorithm to find the roots of a quadratic equation $ax^2 + bx + c = 0$

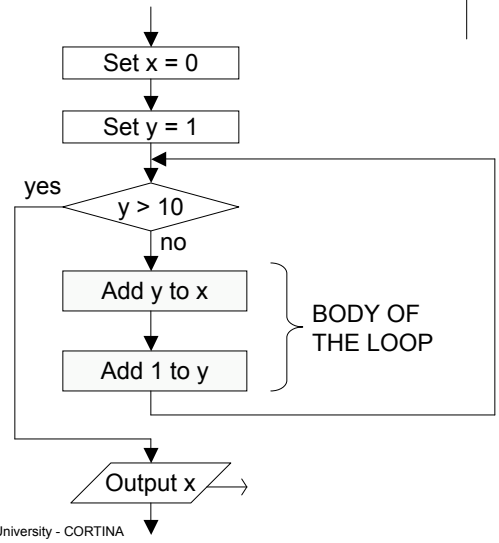


Loop Operations

- A loop operation indicates the sequence of operations that are to be repeated along with a condition to control the number of repetitions
  - Represented in a flowchart using a diamond with a test condition inside
  - Includes one or more flowchart operations with an arrow that flows back to an earlier point in the flow of the algorithm

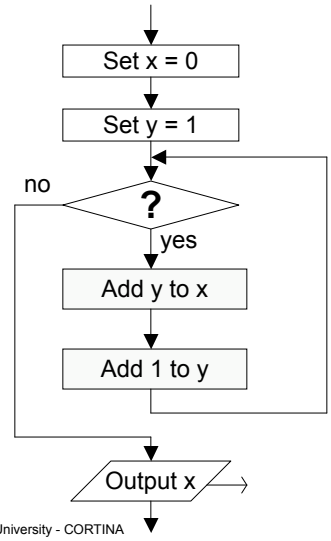
Loop Operations
(cont'd)

- Example:



Loop Operations
(cont'd)

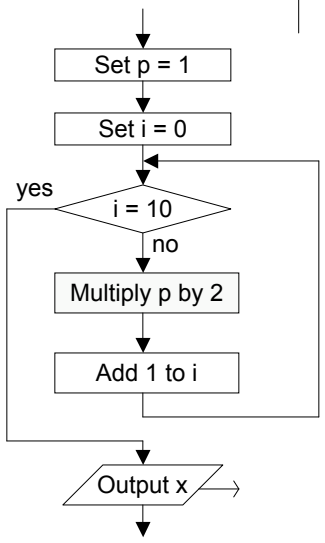
- Example:



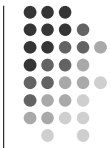
Loop Operations
(cont'd)

- Example:
  Set p = 1
  Do the following 10 times:
  Multiply p by 2
  Output p

  Add an additional unused variable to control the number of times the loop repeats.

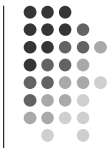


Bubble Sort (Original Algorithm)



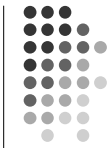
1. Input n
2. Input a vector of values A[0], A[1], …, A[n-1]
3. Do the following n times:
    a. Let i = 0
    b. Do the following n-1 times:
        i. If A[i] > A[i+1], exchange A[i] and A[i+1]
        ii. Add 1 to i
4. Output A

Bubble Sort (Original Algorithm)



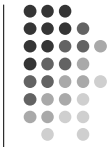
1. Input n
2. Input a vector of values A[0], A[1], …, A[n-1]
3. Let x = 0
4. Do the following while x is not equal to n:
    a. Let i = 0
    b. Do the following n-1 times:
        i. If A[i] > A[i+1], exchange A[i] and A[i+1]
        ii. Add 1 to i
    c. Add 1 to x
5. Output A

Bubble Sort (Original Algorithm)



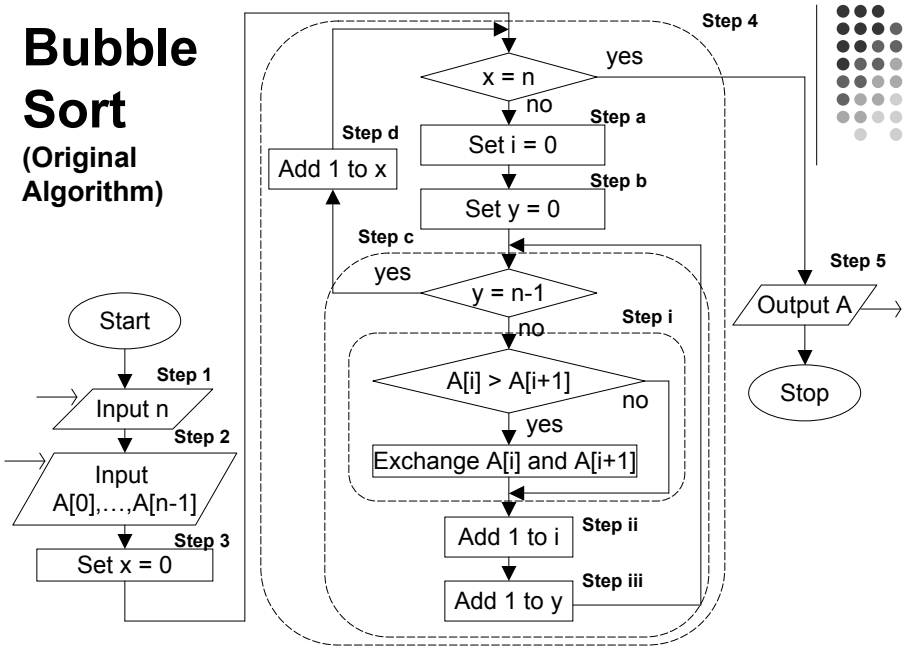
1. Input n
2. Input a vector of values A[0], A[1], …, A[n-1]
3. Let x = 0
4. Do the following while x is not equal to n:
    a. Let i = 0
    b. Do the following n-1 times:
        i. If A[i] > A[i+1], exchange A[i] and A[i+1]
        ii. Add 1 to i
    c. Add 1 to x
5. Output A

Bubble Sort (Original Algorithm)



1. Input n
2. Input a vector of values A[0], A[1], …, A[n-1]
3. Let x = 0
4. Do the following while x is not equal to n:
    a. Let i = 0
    b. Let y = 0
    c. Do the following while y is not equal to n-1:
        i. If A[i] > A[i+1], exchange A[i] and A[i+1]
        ii. Add 1 to i
        iii. Add 1 to y
    d. Add 1 to x
5. Output A

Bubble Sort
(Original Algorithm)



Exercise

Draw a flowchart for the following algorithm:

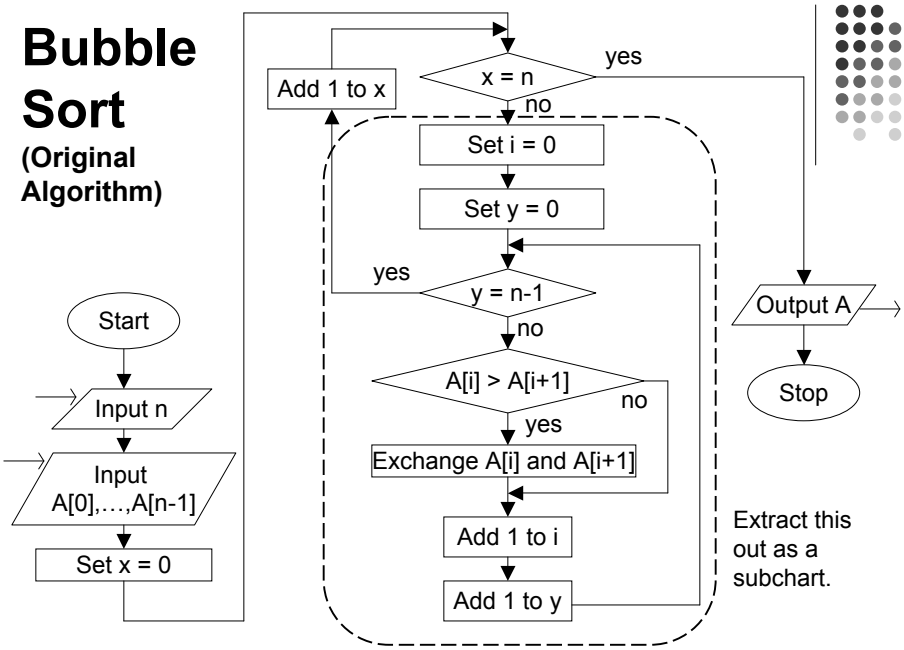
1. Input n
2. Input a vector of values A[0], A[1], …, A[n-1]
3. Set i = 0
4. Set max = A[0]
5. Do the following n-1 times:
   a. Add 1 to i
   b. If A[i] > max, set max = A[i]
6. Output max

Subroutines

- An operation that is executed a number of times at different places in your algorithm can be extracted out and made into a subroutine.
  - A call to a subroutine is represented by a rectangle with a wide arrow to its right
  - A separate flowchart for the subroutine is written in the same way as the initial “main” flowchart that describes the algorithm.

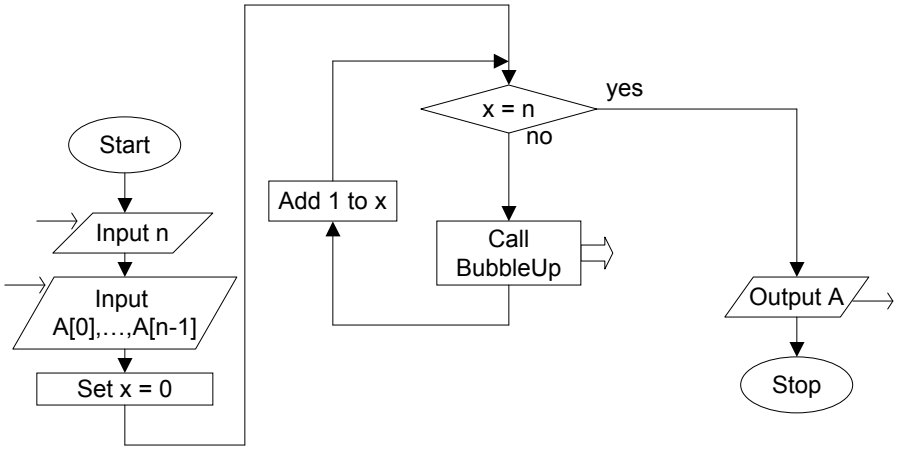
Bubble Sort

(Original Algorithm)



Bubble Sort (Original Algorithm)

Main flowchart



Bubble Sort (Original Algorithm)

Subchart