PART 6 – INTRODUCTION OF ALGORITHM

Dosen Pengampu : Yelna Yuristiary, ST, MT

Outline

- The informal definition of an algorithm
- The operations in an algorithm
- The formal definition of an algorithm

What's an algorithm?

- Dictionary definition: a procedure for solving a mathematical problem in a finite number of steps that frequently involves repetition of an operation; broadly: a step-by-step method for accomplishing some task
- Informal definition: an ordered sequence of instructions (operations) that is guaranteed to solve a specific problem
  - Step 1: Do something
  - Step 2: Do something
  - Step 3: Do something
  - …
  - Step N: Stop, you are finished

Operations

Algorithm is everywhere, not only in CS

Operations in an algorithm

- A sequential operation carries out a single well-defined task. When that task is finished, the algorithm moves on to the next operation.
  - Add 1 cup of butter to the mixture in the bowl
  - Set the value of x to 1
- A conditional operation is the "question-asking" instructions of an algorithm. It asks a question and then select the next operation to be executed according to the question answer
  - If the mixture is too dry, then add 0.5 cup of water to the bowl
  - If x is not equal to 0, then set y equal to 1/x; otherwise, print an error message that says we cannot divide by 0

Variable

carry 5

Set the value of carry to 0 carry 0

I 4

Set the value of I to I+1 I 5 4+1=5
(or I=I+1)
(or add 1 to I)

Operations in an algorithm (Cont.)

- An Iterative operation is a "looping" instruction of an algorithm. It tells us not to go on to the next instruction, but, instead, to go back and repeat the execution of a pervious block of instructions
  - Repeat the previous two operations until the mixture has thickened
  - Repeat steps 1, 2, and 3 until the value of y is equal to +1

Algorithm for programming your VCR

Which instructions are sequential, conditional, and iterative?

Algorithm for Programming Your VCR

Step 1 If the clock and calendar are not correctly set, then go to page 9 of the instruction manual and follow the instructions there before proceeding.

Step 2 Place a blank tape into the VCR tape slot.

Step 3 Repeat steps 4 through 7 for each program that you wish to record, up to a maximum of 10 shows.

Step 4 Enter the channel number that you wish to record, and press the button labeled CHAN.

Step 5 Enter the time that you wish recording to start, and then press the button labeled TIME-START.

Step 6 Enter the time that you wish recording to stop, and then press the button labeled TIME-FINISH.

Step 7 This completes the programming of one show. If you do not wish to record anything else press the button labeled END-PROG.

Step 8 Press the button labeled TIMER. Your VCR is now ready to record.

Formal definition of an algorithm

- 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.
- An algorithm is any well-defined computational procedure that takes some inputs and produce some outputs.
  - An algorithm is a sequence of computation steps that transform the input into the output.
- Sorting problem
  - Input: A sequence of n numbers $< a_1, a_2, \ldots, a_n >$
  - Output: A permutation (reordering) $< a'_1, a'_2, \ldots, a'_n >$ of the input sequence such that $a'_1 \leq a'_2 \leq \ldots \leq a'_n$
  - An instance of the sorting problem
    - (31, 41, 59, 26, 41, 58) ➔ (26, 31, 41, 41, 58, 59)

Is the following an algorithm?

- Step 1: Wet hair
- Step 2: Lather
- Step 3: Rinse
- Step 4: Repeat

...a well-ordered collection...

- Clear and un-ambiguous ordering to these operations
  - What's the next operation when we finish any one operation?
  - At step 4, what operations should be repeated?
- Ambiguous ordering
  - Go back and do it again
    - Go back to step 3 and begin execution from that point
  - Start over
    - Start over from step 1
  - If you understand this material, you may skip ahead
    - If you understand this material, skip ahead to line 21

…of unambiguous and effectively computable operations…

- An unambiguous operation is one that can be understood and carried out directly by the computing agent without needing to be further simplified or explained
  - Primitive (operation)
- Effectively computable or doable operation
  - There exists a computational process that allows the computing agent to complete that operation successfully

…of unambiguous and effectively computable operations… (Cont.)

- Step 1: Make the crust
- Step 2: Make the cherry filling
- Step 3: Pour the filling into the crust
- Step 4: Bake at 350°F for 45 minutes

- Step 1: Make the crust
  - 1.1 Take one and one-third cups flour
  - 1.2 Sift the flour
  - 1.3 Mix the sifted flour with one-half cup butter and one-fourth cup water
  - Roll into two 9-inch pie crusts
- Step 2: make the cherry filling
  - 2.1 open a 16-ounce can of cherry pie filling and pour into the bowl
  - 2.2 add a dash of cinnamon and nutmeg, and stir

Algorithm for making a cherry pie

…of unambiguous and effectively computable operations… (Cont.)

- Which of the following are primitive operations for a computer?
  - Add x and y to get the sum z
  - See whether x is greater than, equal to, or less than y
  - Sort a list of names into alphabetical order
  - Factor an arbitrary integer into all of its prime factors
  - Make a cherry pie

…of unambiguous and effectively computable operations… (Cont.)

- Find and Print out the 100th prime number
  - Step 1: generate a list L of all the prime numbers
  - Step 2: Sort the list L into ascending order
  - Step 3: Print out the 100^{th} element in the list
  - Step 4: Stop

- Write out the exact decimal value of π
  - π cannot be represented exactly
- Set *Average* to *Sum / Number*
  - What if number = 0
- Set the value of result to $\sqrt{N}$
  - What if N < 0
- Add 1 to the current value of x
  - What if x currently has no value

… that produces a result…

- In order to know whether a solution is correct, an algorithm must produce a result that is observable to a user
  - What are the results of the VCR algorithm, cherry-pie making algorithm?
  - Sometimes it is not possible for an algorithm to produce the correct answer because for a given set of input, a correct answer does not exist
    - Error messages (result) should be produced instead

… and halts in a finite amount of time…

- The result must be produced after the execution of a finite number of operations, and we must guarantee that algorithm eventually reaches a statement that says "Stop, you are done"
  - The original shampooing algorithm does not stop

Algorithm for shampooing your hair

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

Alternative algorithm for shampooing your hair

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

What kinds of problems are solved by algorithms?

- Internet routing: single-source shortest paths
- Search engine: string matching
- Public-key cryptography and digital signatures: number-theoretic algorithms
- Allocate scarce resources in the most beneficial way: linear programming
- …

Can every problem be solved algorithmically?

- There are problems for which no generalized algorithmic solution can possibly exist (unsolvable)
- There are also problems for which no efficient solution is known: NP-Complete problem
  - It is unknown if efficient algorithms exist for NP-complete problems
  - If an efficient algorithm exists for any one of them, then efficient algorithms exist for all of them
  - An example: Traveling-Salesman Problem (TSP)
- There are problems that we don't know how to solve algorithmically

Algorithms and other technologies

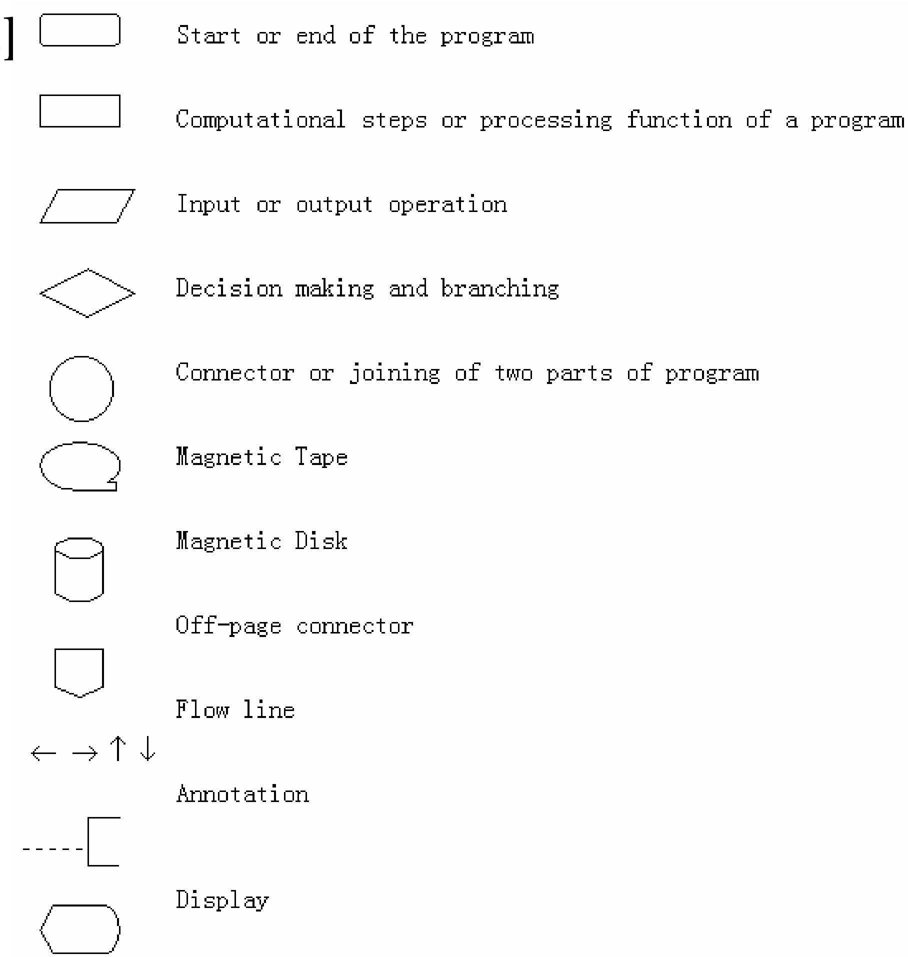
- Algorithms are at the core of most technologies used in contemporary computers
  - The hardware design use algorithms
  - The design of any GUI relies on algorithms
  - Routing in networks relies heavily on algorithms
  - Compilers, interpreters, or assemblers make extensive use of algorithms

Form of Algorithm

- Two form of Algorithm :
  - Flowchart
  - Pseudo Code
- Flowchart
  - A graphical or symbolic representation of a process.
  - Each step in the process is represented by a different symbol and contains a short description of the process step.
  - The flow chart symbols are linked together with arrows showing the process flow direction.

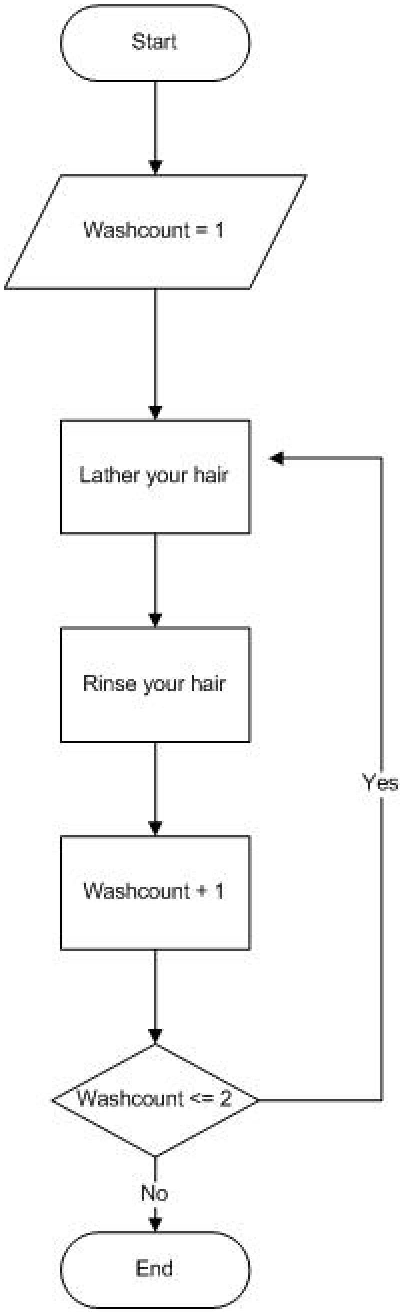
Form of Algorithm Cont'd

- Flowchart Symbol



Form of Algorithm Cont'd

- Flowchart for Shampooing Algorithm



Form of Algorithm Cont'd

- Pseudocode
  - a compact and informal high-level description of a computer programming algorithm that uses the structural conventions of a programming language, but is intended for human reading rather than machine reading.
- Rules for Pseudocode
  - Write only one statement per line
    - Each statement in your pseudocode should express just one action for the computer.

```
PSEUDOCODE:
READ name, hourlyRate, hoursWorked, deductionRate
grossPay = hourlyRate * hoursWorked
deduction = grossPay * deductionRate
netPay = grossPay – deduction
WRITE name, grossPay, deduction, netPay
```

Form of Algorithm Cont'd

- Capitalize initial keyword

READ, WRITE, IF, ELSE, ENDIF, WHILE, ENDWHILE, REPEAT, UNTIL

- Indent to show hierarchy
  - SEQUENCE : keep statements that are "stacked" in sequence all starting in the same column.
  - SELECTION : indent the statements that fall inside the selection structure, but not the keywords that form the selection.
  - LOOPING : indent the statements that fall inside the loop, but not the keywords that form the loop.

```
PSEUDOCODE:
READ name, hourlyRate, hoursWorked
grossPay = hourlyRate * hoursWorked
IF grossPay >= 100
        deduction = grossPay * deductionRate
ELSE
        deduction = 0
ENDIF
netPay = grossPay – deduction
WRITE name, grossPay, deduction, netPay
```

Form of Algorithm

- End Multiline Structures
  - See how the IF/ELSE/ENDIF is constructed above. The ENDIF (or END whatever) always is in line with the IF (or whatever starts the structure).
- Keep statements language independent
  - Resist the urge to write in whatever language you are most comfortable with.

- SELECTION structure of pseudocode :
  - Single IF

```
IF amount < 1000
        interestRate = .06
ELSE
        interestRate = .10
ENDIF
```

Form of Algorithm Cont'd

- Nesting IF

```
READ gameNumber
IF gameNumber = 1
        DO Solitaire
ELSE
        IF gameNumber = 2
                DO Doom
        ELSE
                DO Monopoly
        ENDIF
ENDIF
```

- LOOPING structure
  - WHILE statement

```
count = 0
WHILE count < 10
        ADD 1 to count
        WRITE count
ENDWHILE
WRITE "The end"
```

Form of Algorithm

- REPEAT statement

```
count = 0
REPEAT
        ADD 1 to count
        WRITE count
UNTIL count >= 10
WRITE "The end"
```