2018-23-08 Introduction to Algorithms

Thursday, August 23, 2018 1:18 PM

Four core elements of any algorithm

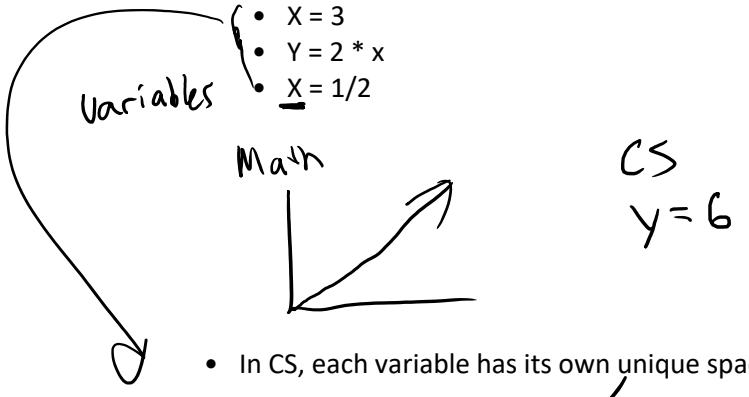
- Command - do something (write something, say something, store something, compute something)
  - Examples:
    - 3 + 4
    - PRINT "Hello, World"
    - Let x = 3
    - Let y = 3 * x

3 assignment operations (different than math equation)

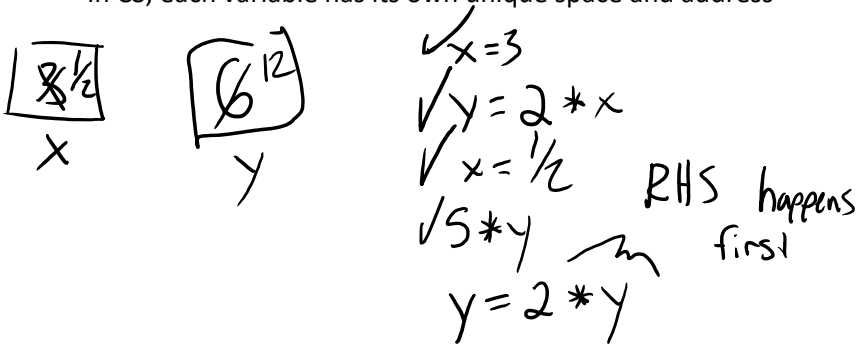
- In CS, y = 9 and will always be 9 until the next assignment

More math vs CS

- X = 3
- Y = 2 * x
- X = 1/2

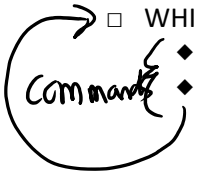


- In CS, each variable has its own unique space and address



- Element #2: Condition
  - Conditions allow for commands to be executed (performed) only under certain conditions
  - Often, conditions are in the format IF <something> THEN <commands>
  - In CS <something> must evaluate to a TRUE of FALSE statement
    - Called Boolean logic
  - Example: driving a car
    - Push pedal to go
    - IF <stop light is red> THEN push brake

        - IF <stop light is green> THEN push pedal
    - Often, conditions "cascade"
        - IF <condition 1> THEN
        - ELSE IF <condition 2> THEN
        - ELSE IF <condition 3> THEN
        - …
        - ELSE THEN
    - Example:
        - IF [time of day < 12 PM] THEN Say "Good Morning"
        - ELSE IF [time of day > 12 PM AND time of day < 5 PM] THEN Say "Good Afternoon"
        - ELSE IF [time of day > 5 PM] THEN say "Good Evening"
        - ELSE THEN Say "I don't know what's happening"
- Element #3: Repetition (loops)
    - Execute one or more commands one or more times
    - Loops are usually time or count dependent
        - Do this for 1 second
        - Do this 100 times
    - The purest loop form is called a WHILE loop
        - Format: WHILE <condition> DO <commands>
        - Example: 99 bottles of beer
            - Let bottle_count = 99
            - WHILE bottle_count > -1
                - PRINT bottle_count # of beers on the wall
                - bottle_count = bottle_count - 1



- Element #4: Abstraction
    - Idea: Remove trivial or unimportant commands from view in order to allow us to focus on the task at hand.
        - Example: Giving someone driving directions to campus. We would prefer to focus on streets, routes, and parking locations.
    - Alternatively, we want to package a common set of commands so that we can easily perform those commands at a later time.
        - Example: If you know the process of mixing pancake batter (put stuff in bowl, add water, mix until ready)
        - This process can be reapplied to the acts of making waffle batter, brownie batter, cake batter, etc.

Algorithmic Examples

- Given a number, write a program to determine whether or not it is odd or even

Solution #1 (turns out wrong!)

1. Let x = some number obtained from the user
2. IF x / 2 is a whole number THEN even
3. ELSE THEN odd

Solution #2

1. Let x = some number obtained from the user
2. IF x < 0 THEN x = -1 * x
3. WHILE x > 1
    a. X = x - 2

4. IF x equals 1 THEN odd
5. ELSE THEN even

Solution #3

1. Let x = some number obtained from the user
2. IF x MOD 2 equals 0 THEN even
3. ELSE odd

Example #2

- Given a sequence of whole numbers (integers), determine if a number denoted by the variable X exists within a given sequence

1. Let numbers = some sequence of integers (e.g. {1, 2, 5, 3, -1, 7, 12})
2. Let x= the number to find (7)