Algorithmic Thinking

FIT 100

To be effective computer users it is necessary to have a general idea how to make a computer solve a problem. Thinking algorithmically is a necessary first step towards solving a problem by computer.

An Assistant's Assistance

- An algorithm is a systematic method for deterministically producing a specified result
- Two participants --
  - The person specifying the algorithm is a *programmer*
  - Some other agent (person or computer) will *execute* the algorithm, i.e. follow its instructions, without intervention of the programmer
- Recipes are an example of algorithms written by chefs and followed by cooks to produce a specified food

S'mores: Place a toasted marshmallow on a Graham cracker and then place a square of chocolate on top

The 5 Properties of Algorithms

- All algorithms must have certain properties if the agent is to execute them successfully without intervention by the programmer
  - Input specified
  - Output specified
  - Definiteness
  - Effectiveness
  - Finiteness

Input/Output Specified

- The "input" is the data that will be transformed by the algorithm to create the output
- In giving an algorithm, state
  - The type of data expected -- whole numbers, letter strings
  - The number of data items expected
  - The structure, if any, of the data expected -- a list, table, etc.
- The "output" is the result of the computation -- its description often forms the name of the algorithm
- The features specified are the same as for input
  - The types of data forming the result
  - The number of data items forming the result
  - The structure of the result

Definiteness

- An algorithm must be explicit about how to realize the computation
- Definiteness is achieved by giving commands that state unambiguously what to do, in sequence
- The commands may be …
  - Conditional, i.e. require a decision to be made, and so must be explicit about how to respond to all different outcomes
  - Repeated, and so must be explicit about when to stop the repetition

The definiteness property assures that the executing agent will always know what command to perform next

Effectiveness

- Effectiveness assures that the agent can perform the algorithm's operations mechanically without intervention
    - No additional inputs, special talent, clairvoyance, creativity or help from Superman
- Effectiveness is achieved by reducing the task to the primitive operations known to the computer
- Definiteness assures the agent knows what command to perform next; effectiveness assures the agent can accomplish the command

A non-effective command would be "Print the NASDAQ's net change for the next trading day"

Finiteness

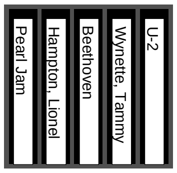
- An algorithm must eventually terminate with either
  - The right output
  - An indication that no solution is possible
- A non-terminating algorithm is useless since it is impossible to distinguish between continued progress and being "stuck"
- Finiteness is relevant to computer algorithms since they typically repeat instructions

```
    3.3
3)10.0000000000…
   9
   10
    9
    1
```

Language

- People write algorithms, but some other agent will execute them … so they must be expressed in some language
- English and other natural languages are poor choices
  - Ambiguous -- multiple interpretations of an utterance "Her lasagna is not very hot"
  - Imprecise -- exact meanings have not been assigned to all words and sentences: *all deliberate speed*
  - Recipes try: *stir* is not a synonym for *fold* or *beat*
- Programming languages are formal notations specifically designed for specifying algorithms
- Visual Basic is the language we'll use

Alphabetize CDs



- *Input*: Unordered CDs filling a slotted rack
- *Output*: CDs in slotted rack, alphabetized

Alphabetizing Algorithms

1 "*Artist_Of*" means the name of the group
2 Pick one end of the rack to be the beginning of the alphabetic sequence. Call that end's slot the "*Alpha*" slot
3 Call the slot adjacent to the *Alpha* slot the "*Bet*" slot
4 If the *Artist_Of* the CD in the *Alpha* slot is later in the alphabet than the *Artist_Of* the CD in the *Bet* slot, then interchange the CDs
5 If there is a slot following the *Bet* slot, begin calling it the "*Bet*" slot and go to step 4; otherwise, continue on
6 If there are two or more slots following the *Alpha* slot, then begin calling the slot following the *Alpha* slot, "*Alpha*" and the slot following it the "*Bet*" slot, and go to step 4; otherwise, stop

Algorithm vs Program

- A program is simply an algorithm specialized to a particular situation …
- Alphabetize CDs is an instance of Exchange Sort
- Exchange Sort can be specialized to other cases
  - Sort CDs by other criteria, e.g. title
  - Sort books by title or other criteria
  - Sort canceled checks, students' homework assignments, vehicles, etc.

The algorithm, being a process with only a limited number of specifics given, is more abstract than is the program