What's an Algorithm?

A problem may be solved using a **parallel** algorithm, essentially a set of algorithmic procedures designed to perform operations concurrently.

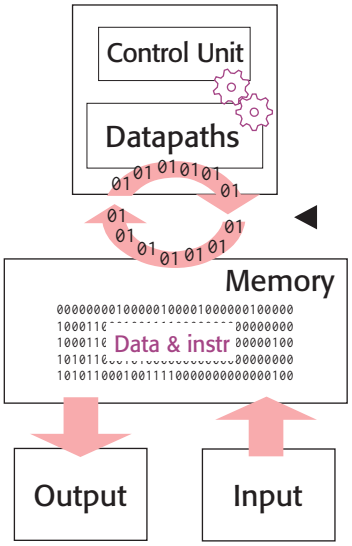
Algorithm ≡ steps to result

- Ordered
- Unambiguous
- Computable
- Terminating

Thinking about the definition

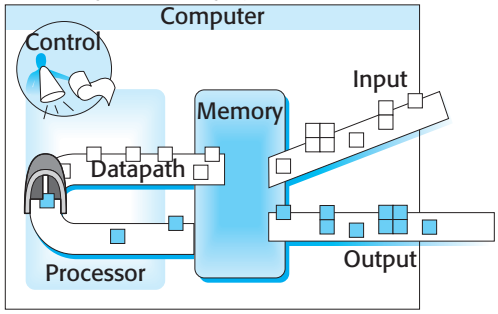
Next time come prepared!

von Neumann Model



Nowadays the most low-end computers have many processors with performance enhancements to speed the flow of instructions and data through these processors.

Computer Organization & Design by Patterson and Hennessy.



© Morgan Kaufmann

Programs and data are stored the same way in the same place.

This model accurately described early modern digital computers. It's still relevant today since most machines use a key concept or another from this model.

A von Neumann machine is meant to execute algorithms

von Neumann Model
The Algorithmic Machine

To run, an algorithm must be expressed in a machine-executable form called **machine language**, or in a machine-readable form called a high-level language which a machine can convert to *its* machine language.

Computer programs are finite sequences of **instruction words** (machine-sepcific bit encodings of operations and operands).

Ordered
Sequencing mechanism (a PC for example)

Unambiguous
Well-defined machine instructions

Computable
Electronic circuits know how (are designed) to execute an instruction in finite time

Terminating
Finite sequences of instruction words

```
00000000100000100001000000100000
10001100010011110000000000000000
10001100010100000000000000000100
10101100010100000000000000000000
10101100010011110000000000000100
```

A MIPS32 program to swap a pair of 32-bit data items in memory.

Specifying the Algorithm

Algorithms determine what the algorithmic machine can do

This is really what the machine needs!

➪ Machine executable

Write algorithm in machine code

Writing machine readable algorithms is what most people call **programming.**

➪ Machine readable

Express the algorithm in machine readable form, let the machine translate to machine code—need a system to describe algorithms to machine

System to express algorithms

Describe algorithm to the machine in efficient, human-friendly yet machine readable form

A 3-step process

❶ Find an algorithmic solution
❷ Describe solution to the machine
❸ Translate to machine code (a compiler)

Step 2 is what some call *coding* the solution.

Programming Past and Future

Programming: a 3-step process

1. Find an algorithmic solution
2. Describe solution to the machine

In the past, limited machines caused more focus on efficient programming systems.

Programming past

"Teaching" machine the algorithm (step 2) used to be critical

2 trends make describing the algorithm (writing a program) less important: 1) more powerful hardware, and 2) automated program generation.

Future

Finding solutions = algorithms (step 1) increasingly more important

Programming Models

A programmer can directly write an algorithmic solution, or use other higher-level problem-solving **abstractions.**

Assembly is a low-level procedural language. FORTRAN, Pascal, and C are examples of high-level procedural languages.

Algorithmic models

Procedural (imperative) programming: sequence, decision, iteration

Beyond algorithms

Object-oriented programming

Functional programming

Logic programming …

Object-oriented programs describe solution (rather than steps) in terms of objects (key players) and their behavior (how they interact).

Smalltalk, C++, and Java are examples of object-oriented languages.

First Exercise

- **Lecture summary: the calendar**
- **Programming exercises: P1**
  - Follow instructions
  - Submit next class
- **Exercise walkthrough**