What is an Algorithm?

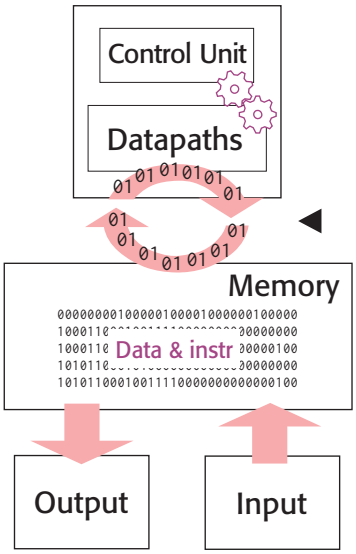
A **parallel** algorithm may solve a problem using a set of algorithmic procedures designed to perform operations concurrently.

Algorithm ≡ steps to result

- Ordered
- Unambiguous
- Computable
- Terminating (halt in finite time)

Algorithms are powerful but not limitless. We learn about limitations of algorithmic solutions later.

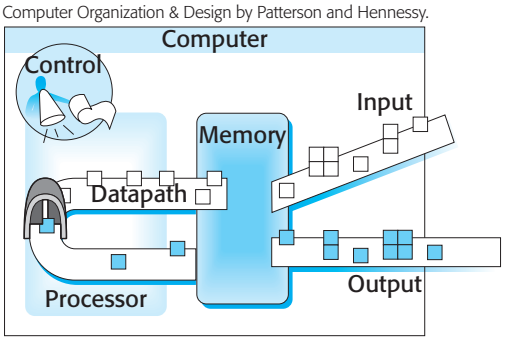
von Neumann Model



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

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

Computer Organization & Design by Patterson and Hennessy.



© Morgan Kaufmann

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.

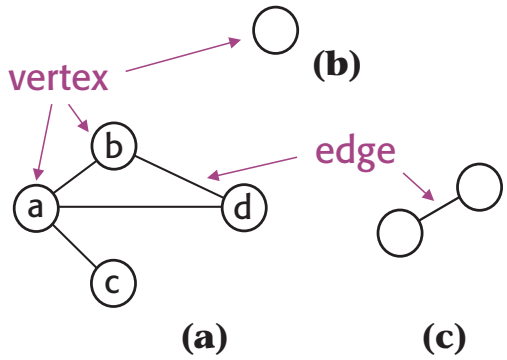
This Week

- Read syllabus (My Course) 
- Sign-up in course group
- Check the programming guide
- Review graphs (next)

Definitions/Terminology

16 key-terms

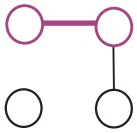
- Graph, vertex, edge
- Adjacent vertex
- Vertex degree



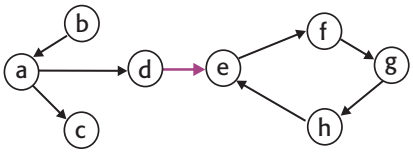
(a) (b) (c)

$$G = \langle V, E \rangle$$
$$V = \{v_i (1 \le i \le n)\}$$
$$E = \{(u, v) | u, v \in V\}$$

Exercise
Write a formal (=math) definition to describe vertex adjacency (Check answers later.)



(d)



(e)

Exercise
Specify graphs (a) and (e) using the definitions.

❷ Edge Properties

- ➪ **Undirected graph**
- ➪ **Directedgraph**
- ➪ **Complete, dense/sparse**

➪ Edge direction

✎ Undirected graph

$$\overset{\text{iff}}{\Longleftrightarrow} \forall (u,v) \in E, (u,v) = (v,u)$$

✎ Directed graph: def, example

Quiz
What's the efficiency of any algorithm that traverses all edges of an undirected graph?

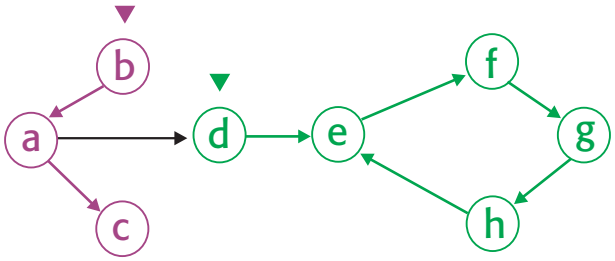
➪ Number of edges (undirected)

Graphs: The Basics

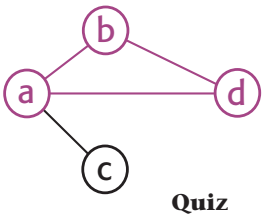
❸ More Definitions

- ➪ **Path, cycle/loop**
- ➪ **Path length**
- ➪ **Weighted graph**

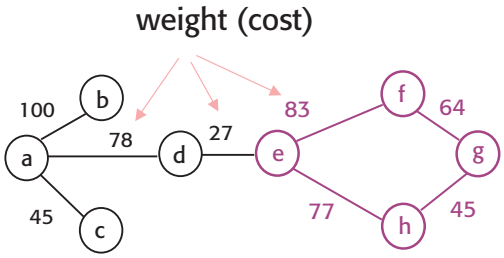
Exercise
Specify the highlighted paths.



Quiz
Identify simple paths giving a reason if not? What's the length in each case?

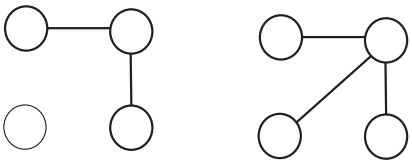


Quiz
What's the smallest possible cycle?

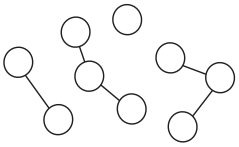


❹ Most Basic Properties

- Connected graph
- Connected component
- Acyclic graph

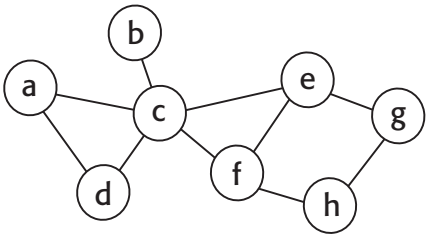


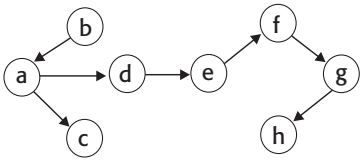
(a) **(b)**



Exercise
How many connected components?

Exercise
List as many cycles as you can. How many are unique cycles?





Directed acyclic graph (DAG)

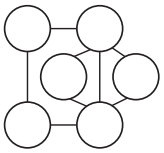
Quiz
Is the DAG connected?

Special Cycles

Quiz
Is a Hamiltonian circuit a simple path? Why?

⇨ Hamiltonian cycle (circuit)

Revisit edges but not vertices



⇨ Eulerian circuit

Reuse vertices but not edges

Exercise
Label vertices and trace the Euler circuit.

https://www.cs.sfu.ca/~ggbaker/zju/math/euler-ham.html

The Data Structure

🕐 Connections: think about it

- ✎ Tree: acyclic connected graph
- ✎ List: graph with one simple path