VGP332 – Artificial Intelligence

Instructor: Peter Chan

Agenda

- Intro & Admin
- Intro to AI
- Math & Physics Review
- Intro to Pathfinding
- Graphs & Nodes
- Breadth-First Search
- Depth-First Search
- Assignment Overview

Agenda

- Intro & Admin
- Intro to AI
- Math & Physics Review
- Intro to Pathfinding
- Graphs & Nodes
- Breadth-First Search
- Depth-First Search
- Assignment Overview

Introductions & Administration

- Introduction
- Course syllabus
- Overview

Agenda

- Intro & Admin
- Intro to AI
- Math & Physics Review
- Intro to Pathfinding
- Graphs & Nodes
- Breadth-First Search
- Depth-First Search
- Assignment Overview

Introduction to AI

- What is Artificial Intelligence?
- What is Artificial Intelligence in Games?
- Is there a difference?

AI

- Computer vision
- Speech recognition
- Language recognition
- Expert systems
- Pathfinding / navigation
- Goal-driven
- …

AI in Games

- Give the illusion of intelligence
- Needs to be *fast*
- Often needs to be suboptimal

AI in Games

- Give the illusion of intelligence
  - Behaviour that players would deem intelligent
  - Can use any means necessary:
    - Extra hit points
    - Scripted sequences
    - Can cheat unobtrusively
- Needs to be *fast*
- Often needs to be suboptimal

AI in Games

- Give the illusion of intelligence
- Needs to be *fast*
  - Limited # of cycles
  - Can't afford expensive algorithms
- Often needs to be suboptimal

AI in Games

- Give the illusion of intelligence
- Needs to be *fast*
- Often needs to be suboptimal
  - Perfect computer player is frustrating
  - Balancing act between too smart & too dumb

AI in Games

The goal of AI in a game is to stop the computer from looking stupid

AI Examples

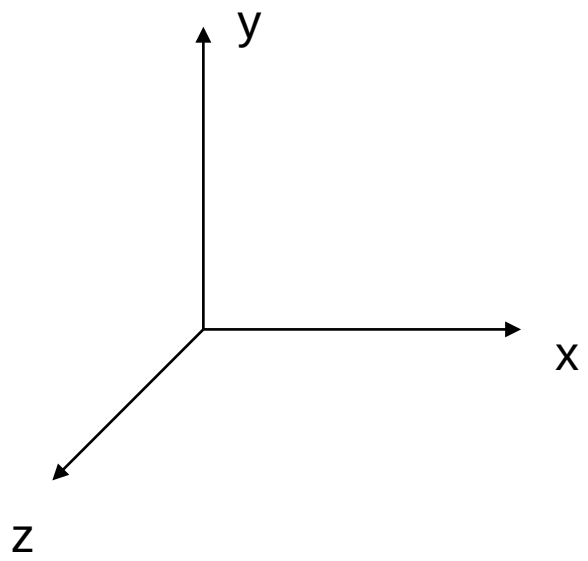
- Can you think of examples of smart AI in games you've played?
- What about examples of bad AI?

Agenda

- Intro & Admin
- Intro to AI
- Math & Physics Review
- Intro to Pathfinding
- Graphs & Nodes
- Breadth-First Search
- Depth-First Search
- Assignment Overview

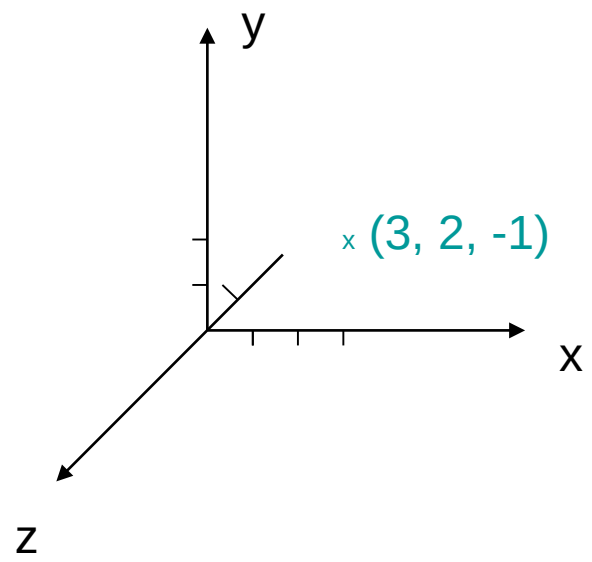
Math & Physics Review

Cartesian Coordinates



- 1 dimensional = 1 axis (x)
- 2 dimensional = 2 axes (x, y)
- 3 dimensional = 3 axes (x, y, z)

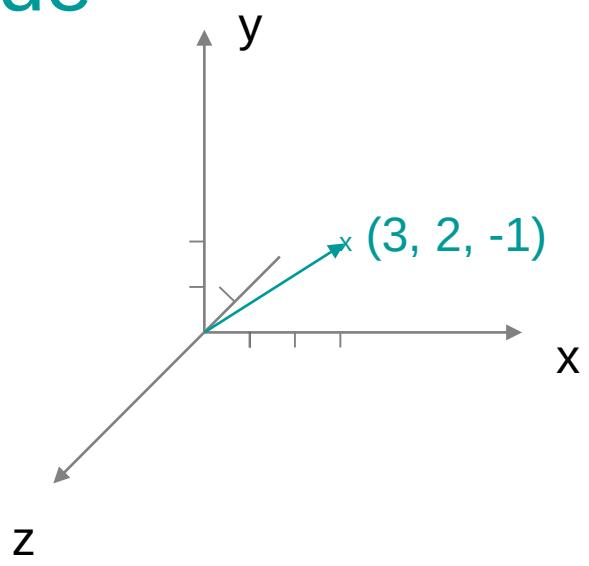
Point



- A point in 3D space has
  3 values for each of the 3 axes

Vector

- Has direction and magnitude
- With (x, y, z) values, vector points from origin to (x, y, z)



Vector Math

- Addition piecewise by axis
  v + **w** = (**v**.x + **w**.x, **v**.y + **w**.y, **v**.z + **w**.z)

- Subtraction is identical
  v - **w** = (**v**.x - **w**.x, **v**.y - **w**.y, **v**.z - **w**.z)

- Commutative:

  v + w = w + v

Vector Math

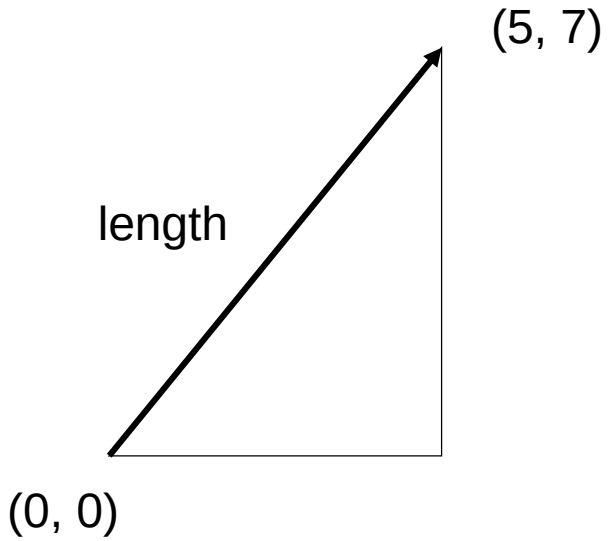
- Multiplication by scalar
  v * s = (**v**.x * s, **v**.y * s, **v**.z * s)

- Division is identical
  v / s = (**v**.x / s, **v**.y / s, **v**.z / s)

  (Note: watch for division by zero!)

Vector Length

- Length (2D)



Use Pythagoras:

$Length^2 = x^2 + y^2$

$Length = sqrt(x^2 + y^2)$

For example:

Length = sqrt(5^2 + 7^2)

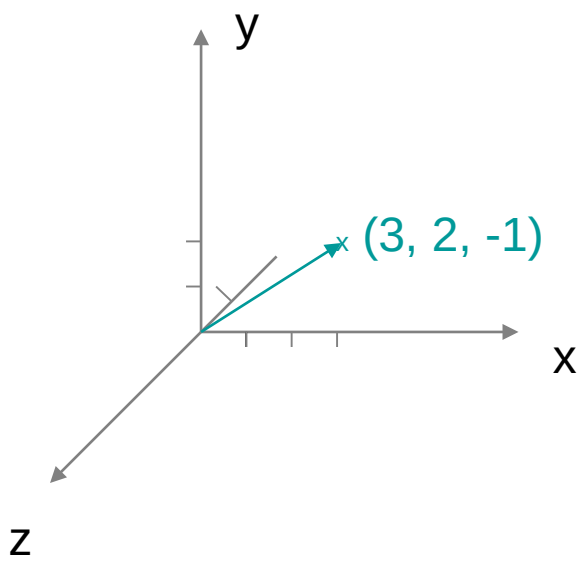
Length = sqrt(25 + 49)

Length = sqrt(74)

Length = 8.602…

Vector Length

- Length (3D)



Use Pythagoras:

$Length^2 = x^2 + y^2 + z^2$

$Length = sqrt(x^2 + y^2 + z^2)$

For example:

Length = sqrt(3^2 + 2^2 + -1^2)

Length = sqrt(9 + 4 + 1)

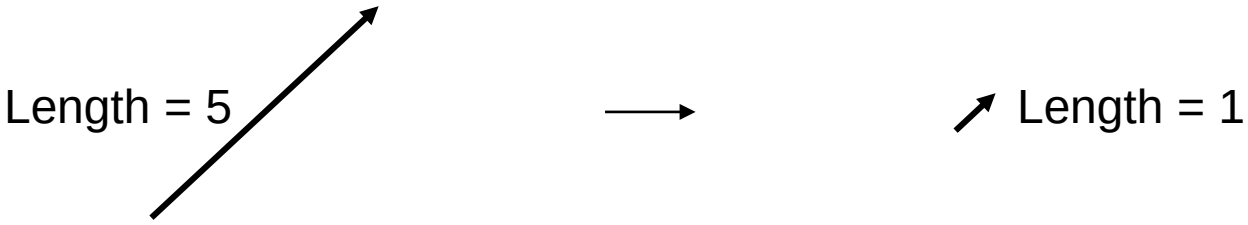
Length = sqrt(14)

Length = 3.742…

Notation: Given a vector **v**, its length is denoted ||**v**||

Vector Normalization

- Normalization:
  - Keep vector direction the same
  - Scale vector magnitude to unit length



Divide vector by its length: $\frac{\mathbf{v}}{\|\mathbf{v}\|}$

Notation: Given a vector $\mathbf{v}$, its unit vector is denoted $\hat{\mathbf{v}}$

Vector Dot Product

- Given two vectors, **v** and **w**
- Defined as:

  $\mathbf{v} \cdot \mathbf{w} = ||\mathbf{v}||\ ||\mathbf{w}|| \cos \Theta$

  (where Θ is the angle between **v** and **w**)

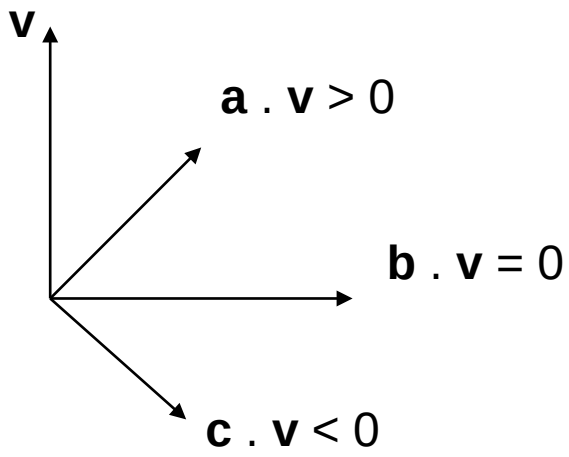
  returns scalar result
- Can be simplified to:

  $\mathbf{v} \cdot \mathbf{w} = \mathbf{v}.x * \mathbf{w}.x + \mathbf{v}.y * \mathbf{w}.y + \mathbf{v}.z * \mathbf{w}.z$
- Commutative:

  $v \cdot w = w \cdot v$

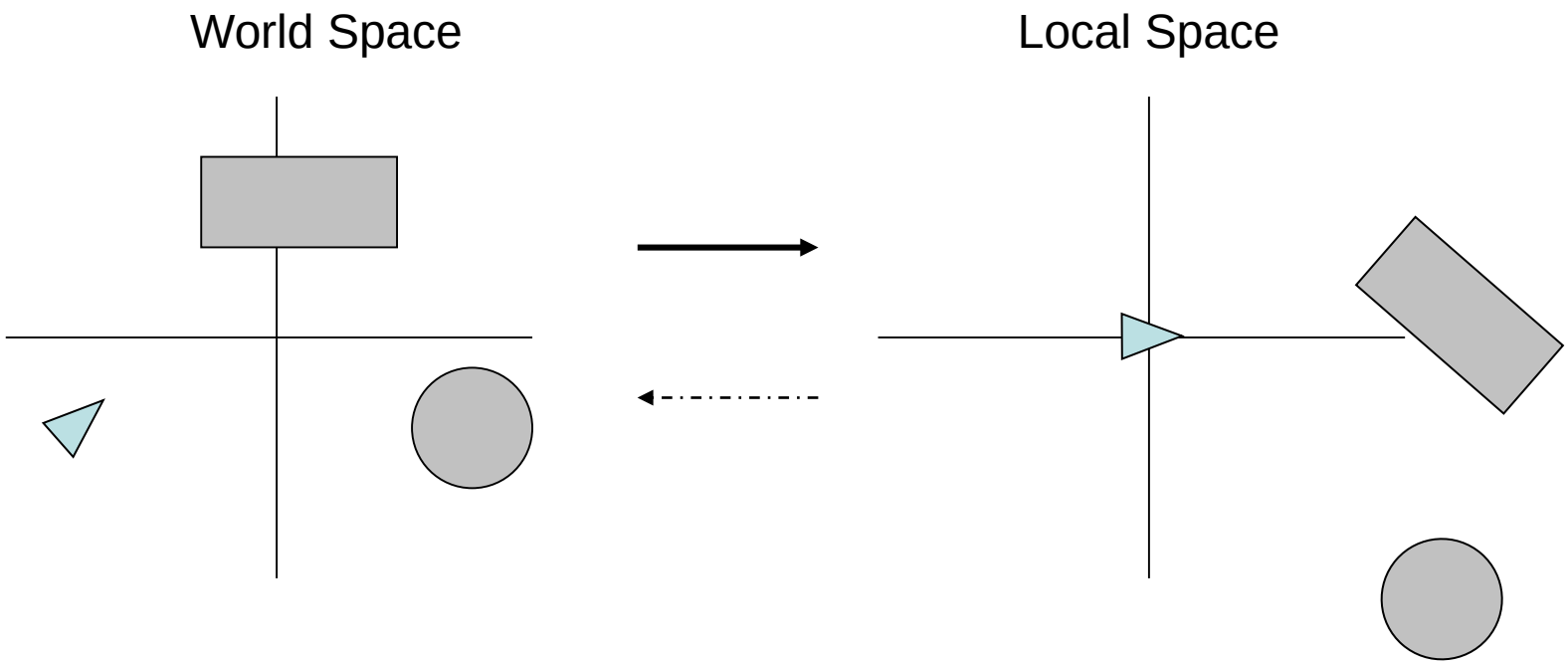
Vector Dot Product

- Determine general angle between vectors



- Quick test to see if target is in front, beside or behind

World Space vs. Local Space



Physics Review

Physics Review

- Velocity = Δ Distance / Δ Time

  $P_{t+1} = P_t + V\Delta t$

- Acceleration = Δ Velocity / Δ Time

  $V_{t+1} = V_t + a\,\Delta t$

- Force = mass x acceleration

Why physics in AI?

- Realistic-looking behaviours & motion
- Useful for prediction

Agenda

- Intro & Admin
- Intro to AI
- Math & Physics Review
- Intro to Pathfinding
- Graphs & Nodes
- Breadth-First Search
- Depth-First Search
- Assignment Overview

Pathfinding

- How would you define pathfinding?

Pathfinding

- Go from point A to point B
- Avoid obstacles
- Avoid getting stuck
- Simpler in 2D, but can extend to 3D
- Find shortest path (optional?)
- Should be dynamic

Pathfinding

- How would you do pathfinding?

Pathfinding is Hard!

http://www.youtube.com/watch?v=lw9G-8gL5o0

Agenda

- Intro & Admin
- Intro to AI
- Math & Physics Review
- Intro to Pathfinding
- Graphs & Nodes
- Breadth-First Search
- Depth-First Search
- Assignment Overview

Graphs

- A graph is formally defined as:

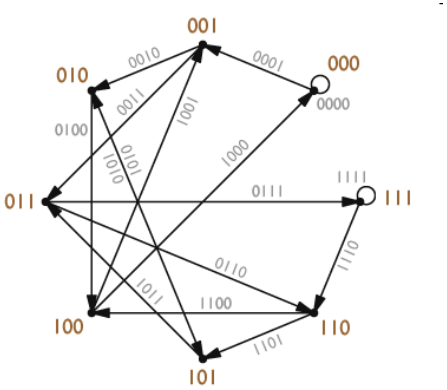
$$G = \{N, E\}$$

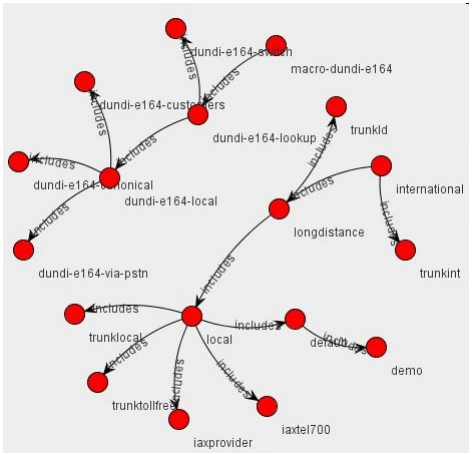
where N is the set of nodes

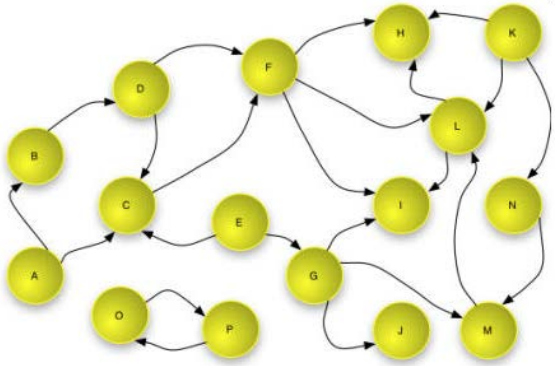
and E is the set of edges connecting those nodes

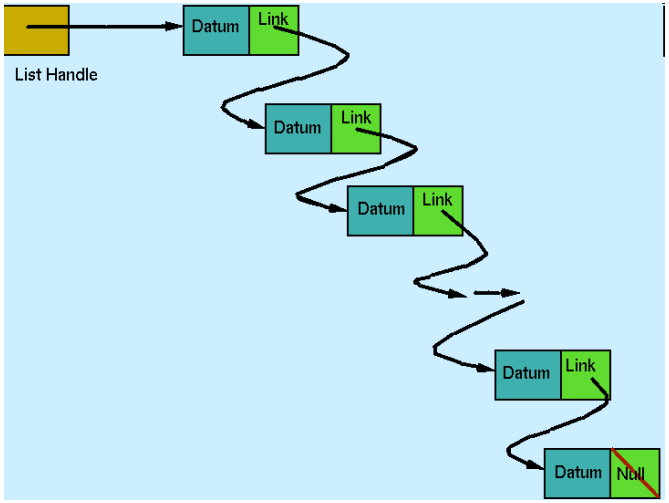
Graphs

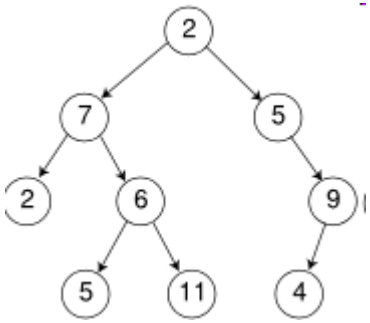
- What does that really mean?

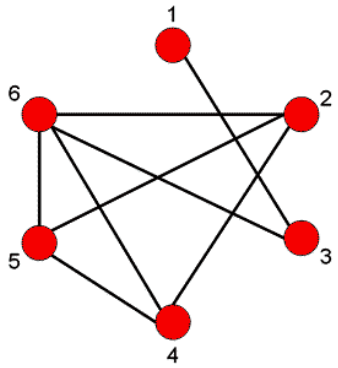






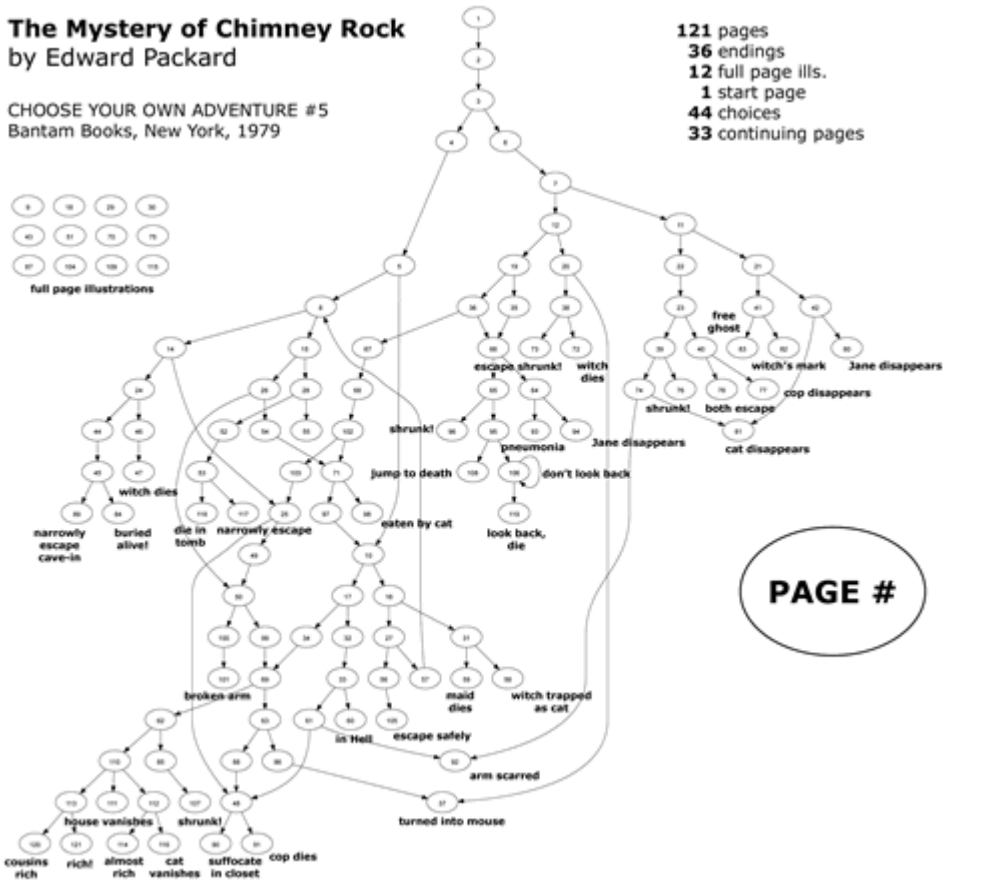






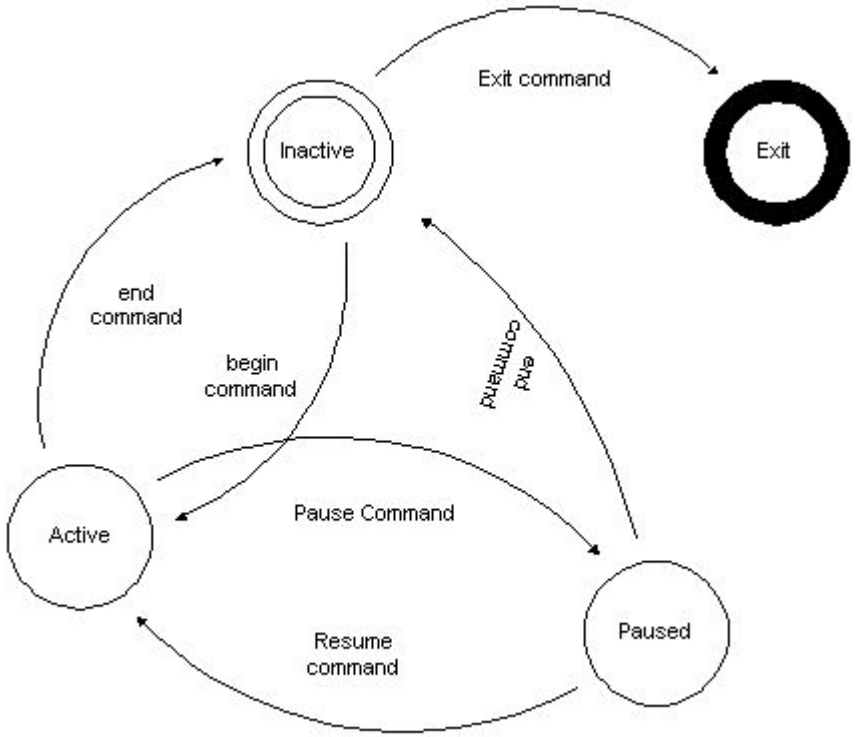
Graph Versatility

- Decisions



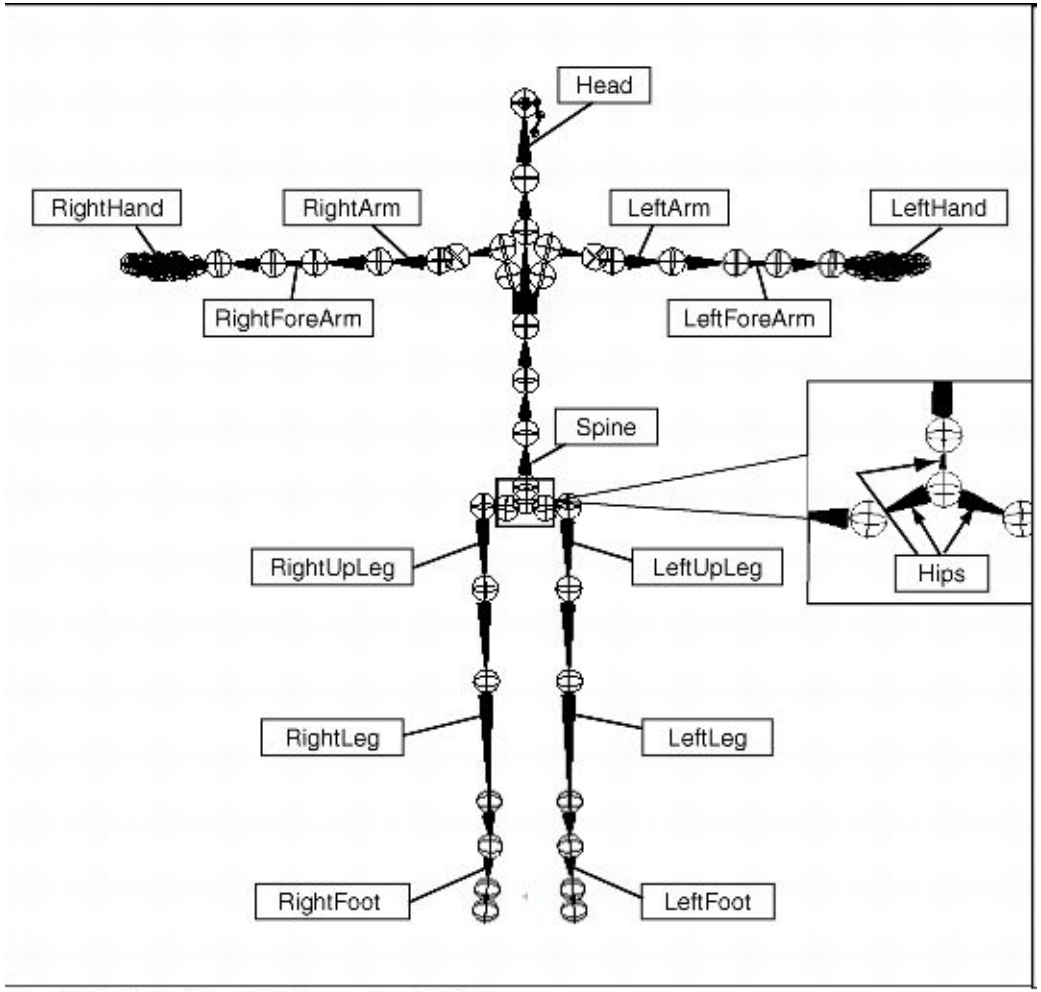
Graph Versatility

- States



Graph Versatility

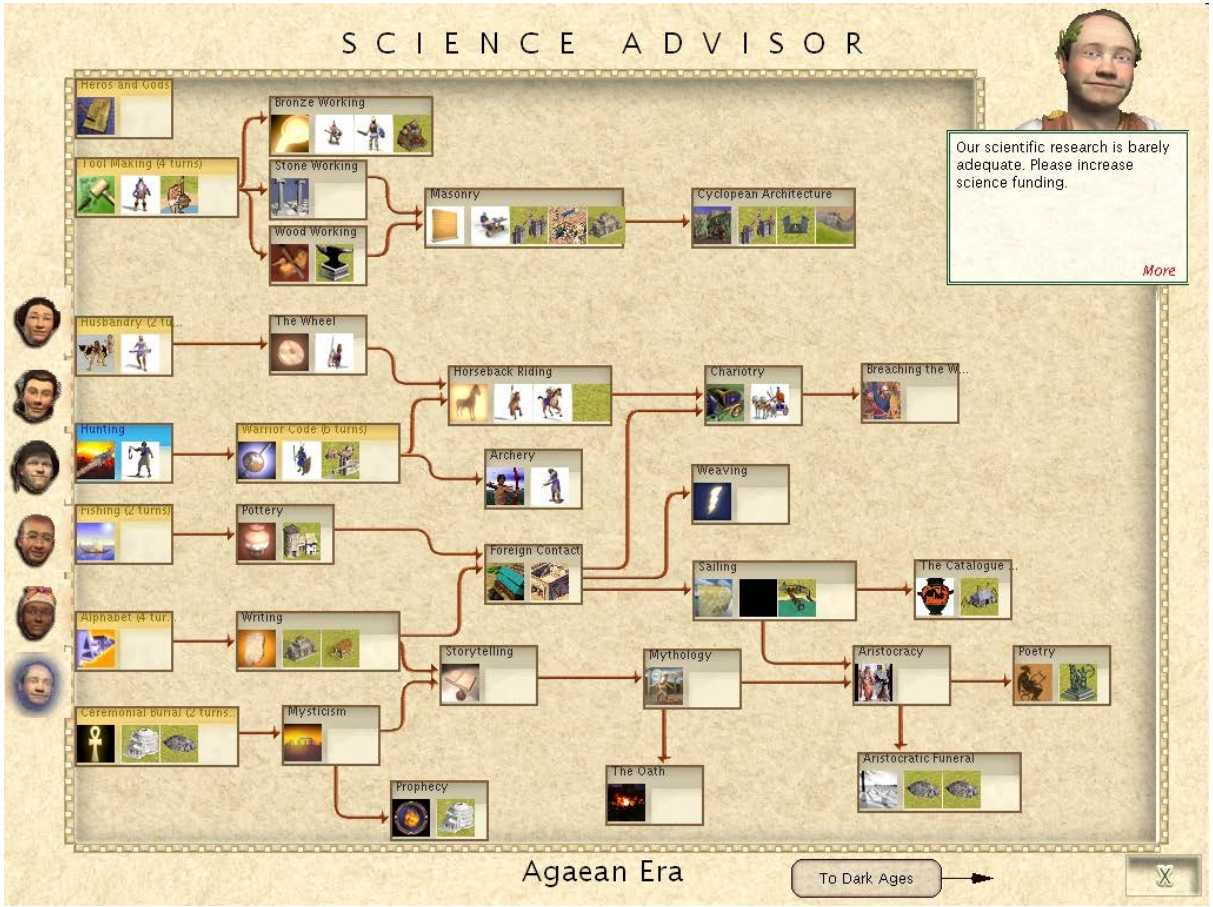
- Geometry



igure 6: Basic skeleton with Base nodes labelled

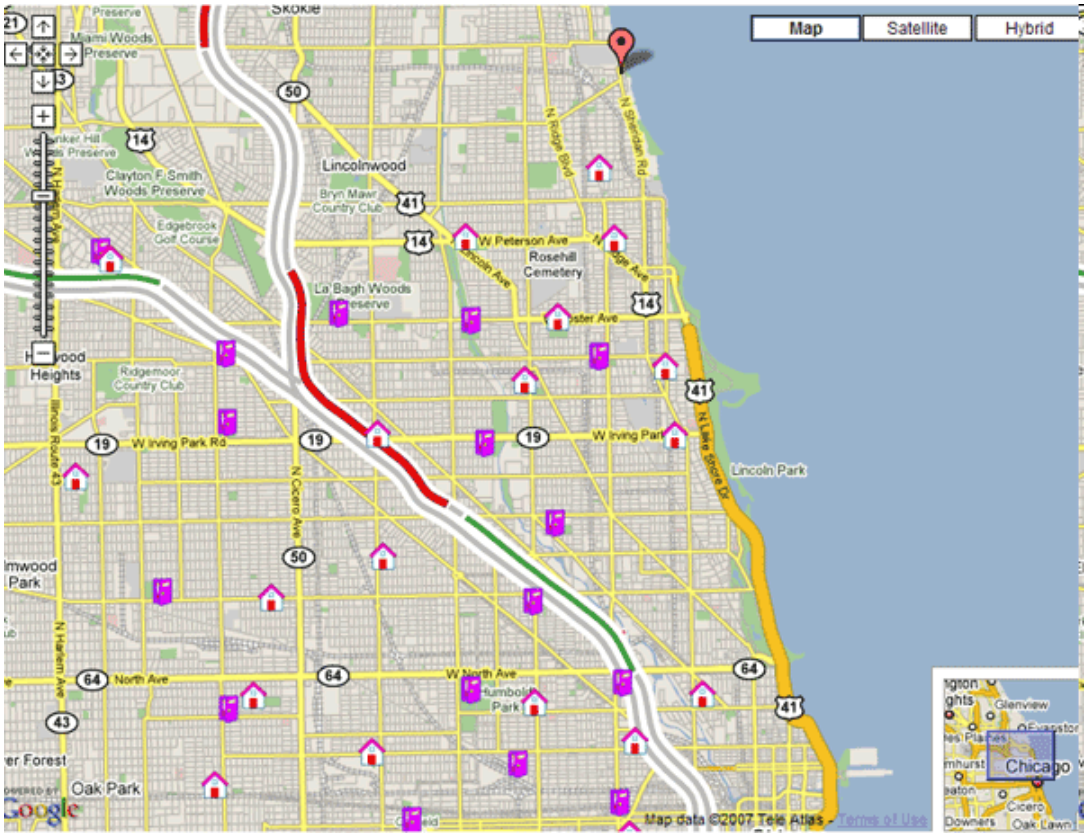
Graph Versatility

- Dependencies (e.g. tech trees)



Graph Versatility

- Navigation



Pathfinding using Graphs

- Warcraft II

Can you think of a way to represent this map using a graph?

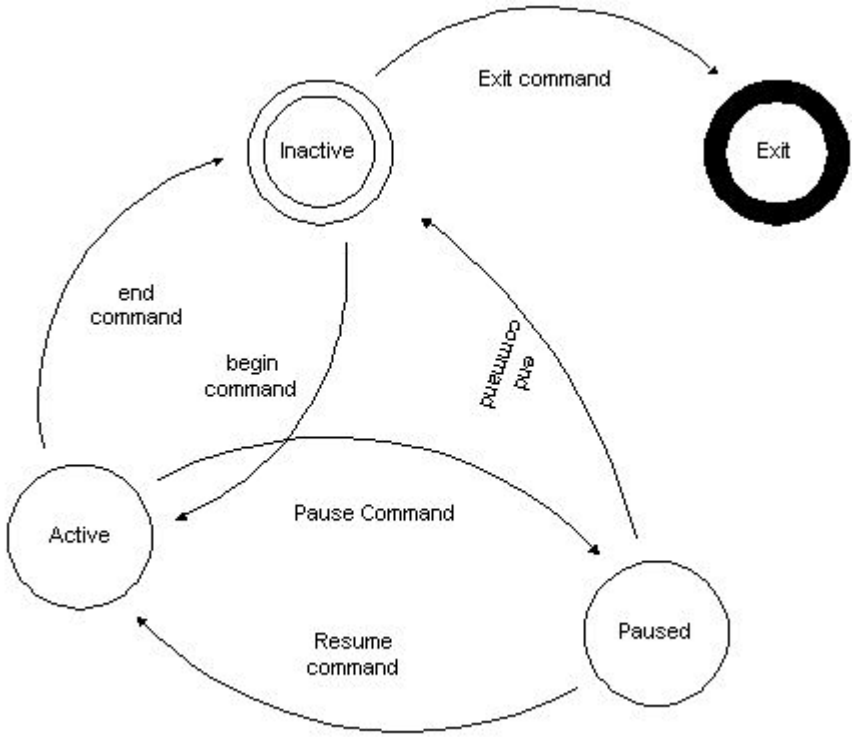
Pathfinding using Graphs

- Warcraft III

What about this one?

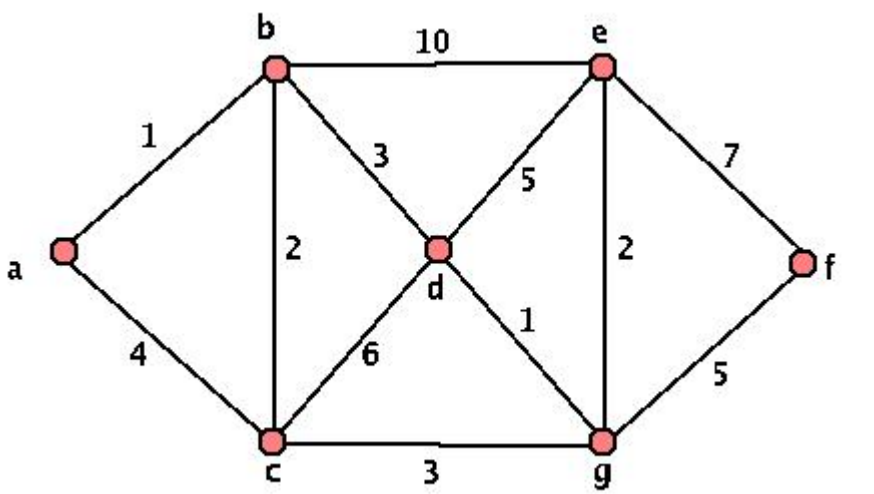
Additional Graph Features

- Directed graphs



Additional Graph Features

- Weighted edges

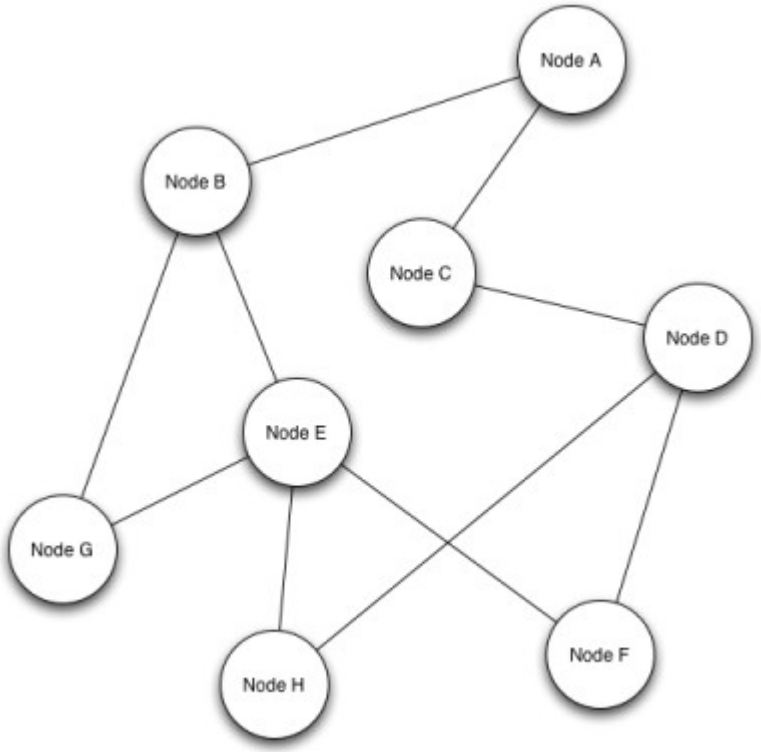


Graph Implementation

- Adjacency matrix
- Adjacency list
- Grid-based

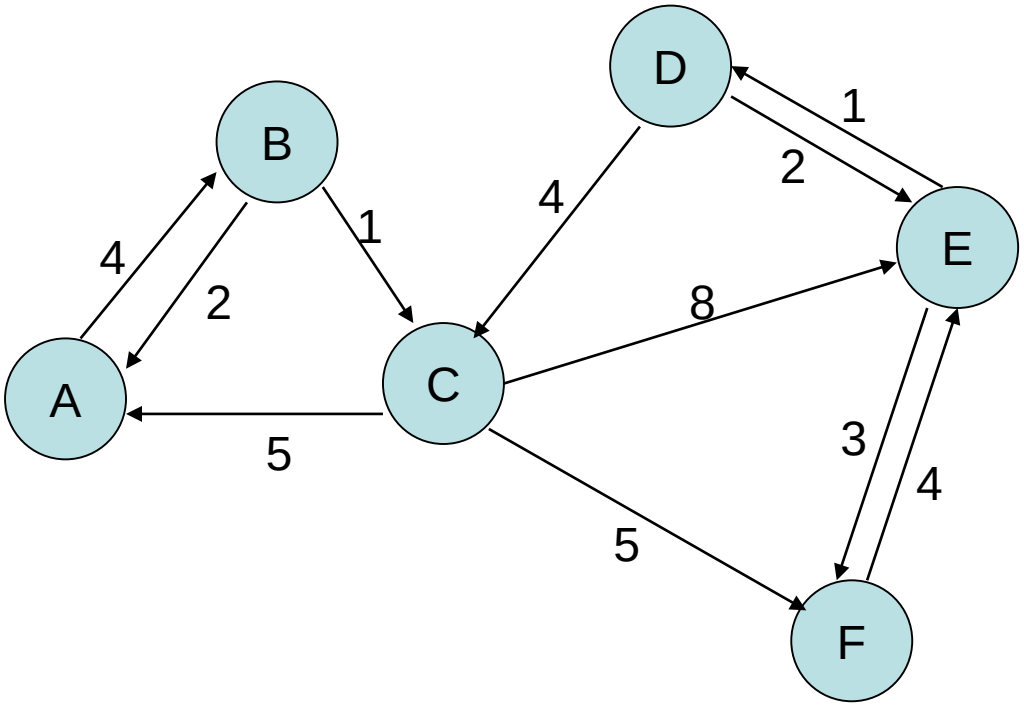
Adjacency Matrix

	A	B	C	D	E	F	G	H
A	0	1	1	∞	∞	∞	∞	∞
B	1	0	∞	∞	1	∞	3	∞
C	1	∞	0	1	∞	∞	∞	∞
D	∞	∞	1	0	∞	2	∞	5
E	∞	1	∞	∞	0	2	1	1
F	∞	∞	∞	2	2	0	∞	∞
G	∞	3	∞	∞	1	∞	0	∞
H	∞	∞	∞	10	1	∞	∞	0



Adjacency Matrix

- Create the adjacency matrix for this graph:

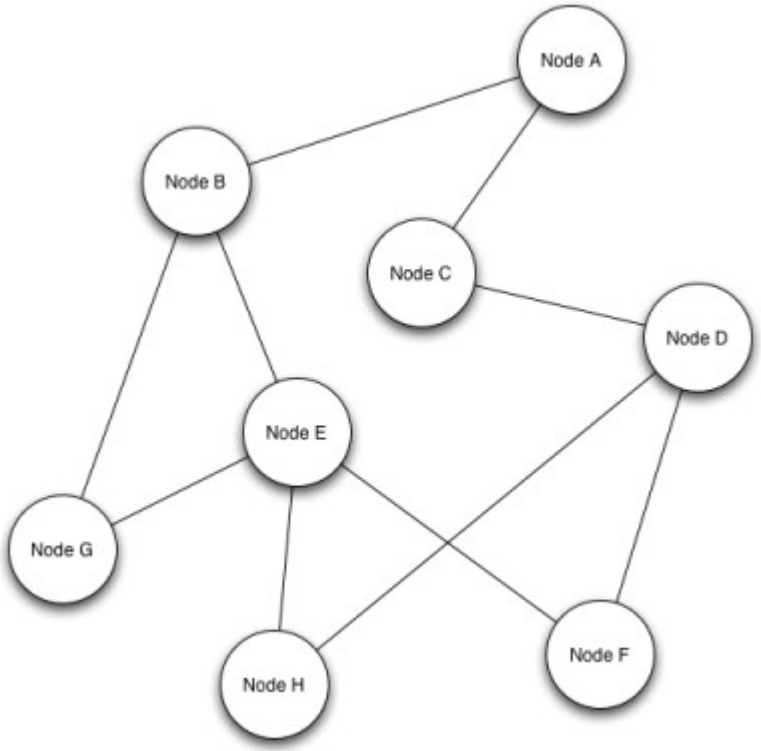


Adjacency Matrix

- Pro: can represent directed graphs easily
- Pro: can represent weighted graphs easily
- Con: lots and lots of wasted space

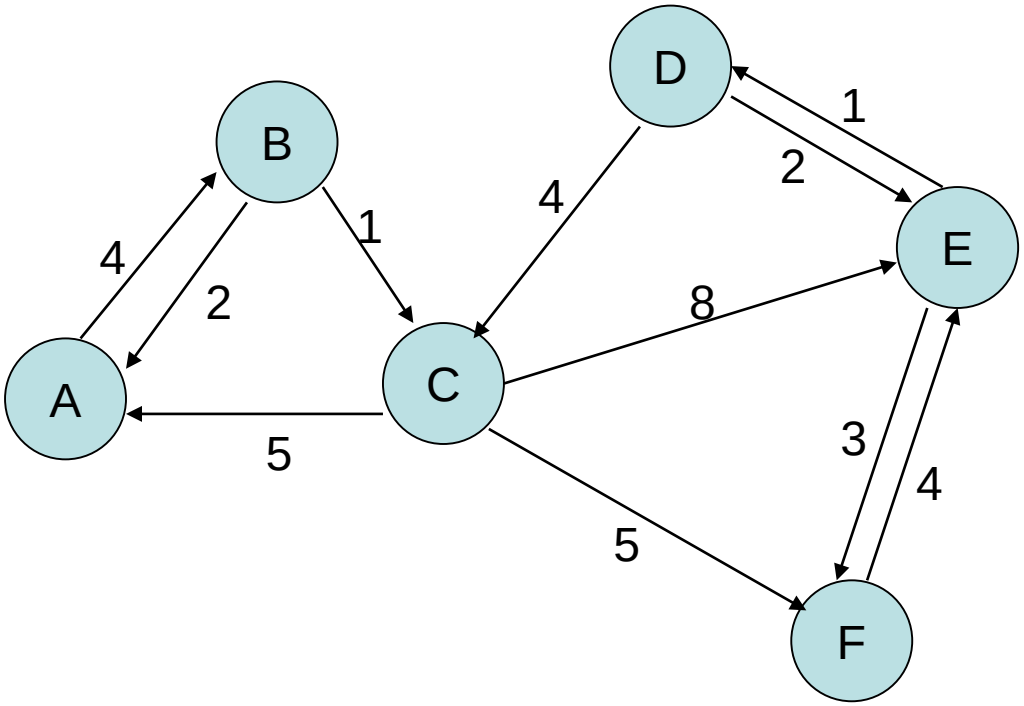
Adjacency List

A	B	C		
B	A	E	G	
C	A	D		
D	C	F	H	
E	B	F	G	H
F	D	E		
G	B	E		
H	D	E		



Adjacency List

- Create the adjacency list for this graph:

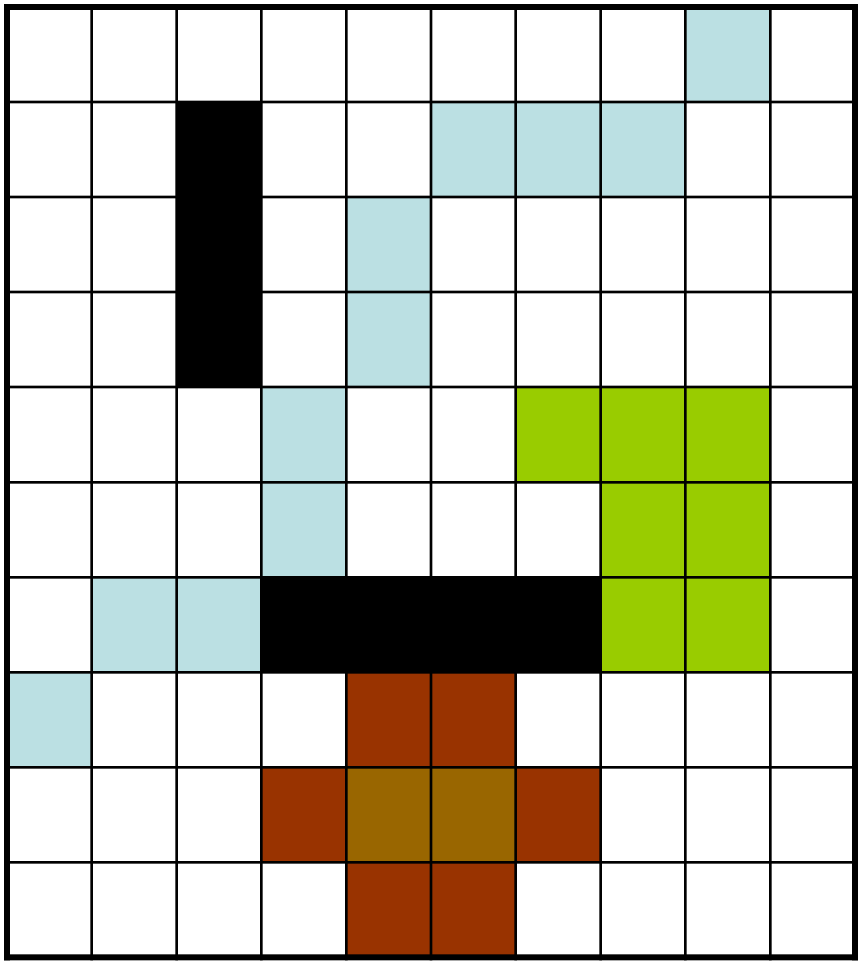


Adjacency List

- Pro: only takes up a little space
- Pro: can represent directed graphs easily
- Con: additional attribute needed for weight

Grid-Based

- Graph is represented implicitly
- Edge weights & directions represented by terrain types
- Pros and cons?



Compare grid-based map

Agenda

- Intro & Admin
- Intro to AI
- Math & Physics Review
- Intro to Pathfinding
- Graphs & Nodes
- Breadth-First Search
- Depth-First Search
- Assignment Overview

Graph Searches

- Used to find desired node
  - Existence
  - Connection
  - State
  - Destination (navigation)

Graph Searches

- Open list:
  - list of nodes you need to consider in the steps ahead
- Closed list:
  - list of nodes you've already visited and don't need to consider again
- Parent:
  - for each node, keep track of which node it was expanded from

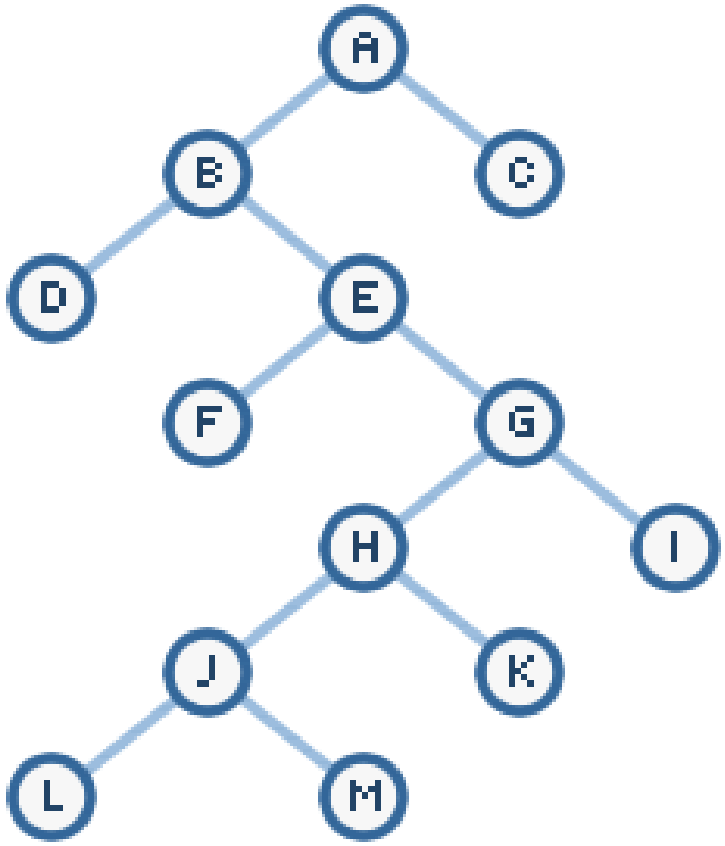
Graph Searches

```
put start node in open list
while end node not reached && open list isn’t empty:
     move node N from open list to closed list
     expand node N:
       if expanded node isn’t in open or closed lists:
             add expanded node to open list
```

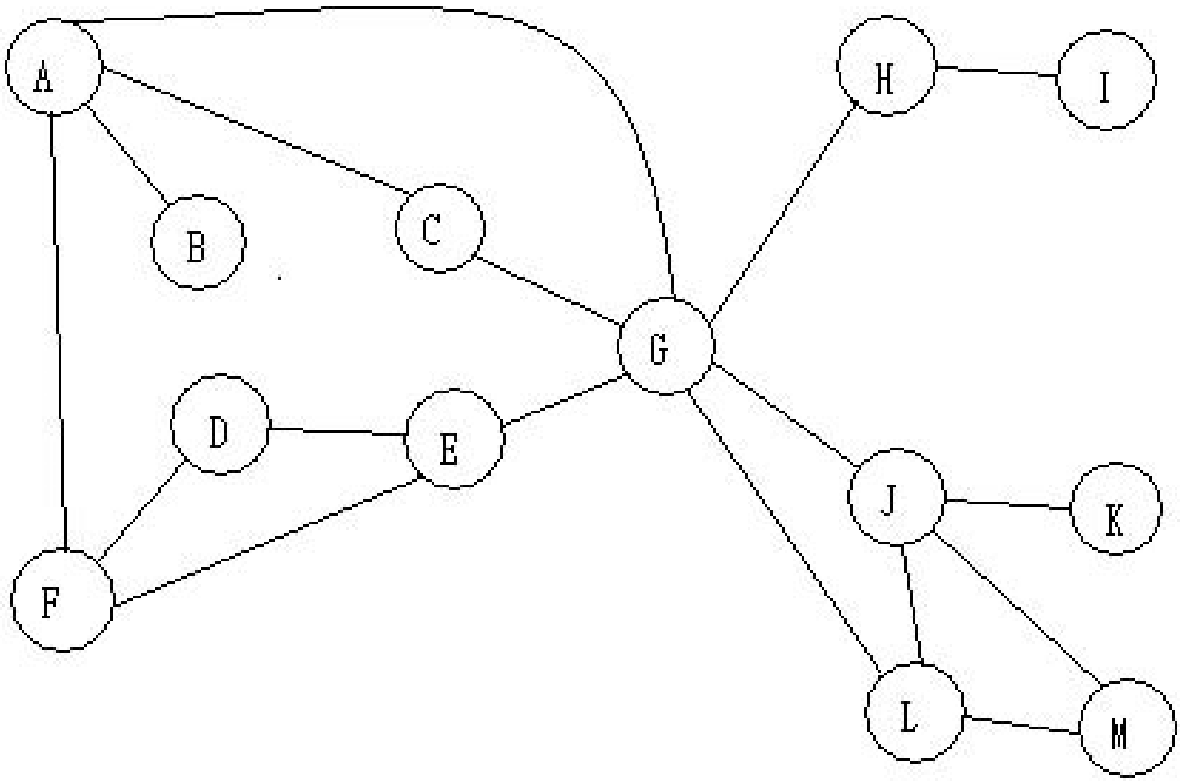
Breadth-first Search

- Uninformed search
- Choose node from front of open list
- Expanded nodes get added to back of open list

BFS Example 1



BFS Example 2



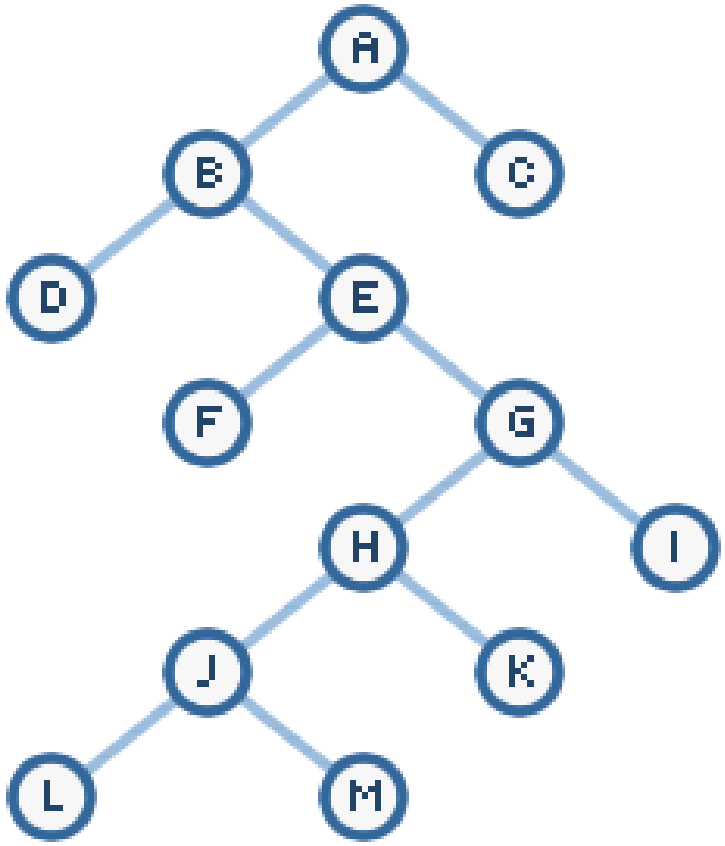
Agenda

- Intro & Admin
- Intro to AI
- Math & Physics Review
- Intro to Pathfinding
- Graphs & Nodes
- Breadth-First Search
- Depth-First Search
- Assignment Overview

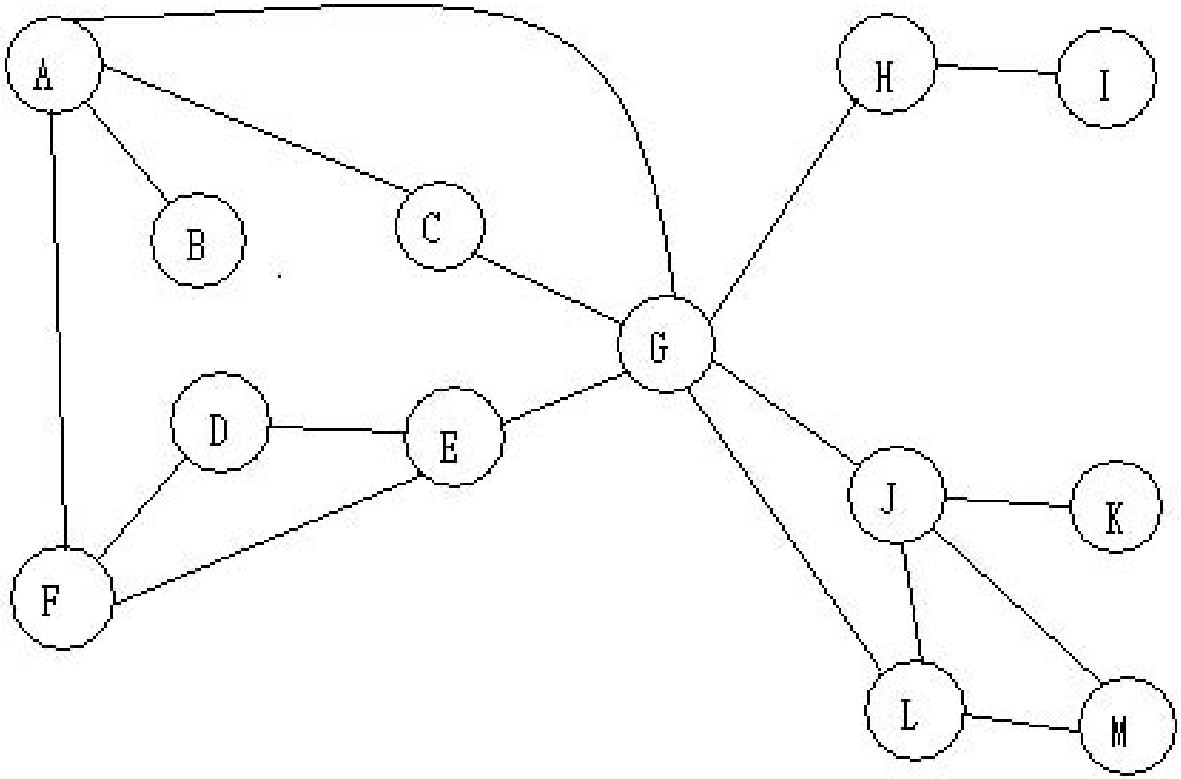
Depth-first Search

- Uninformed
- Choose node from front of open list
- Expanded nodes get added to front of open list

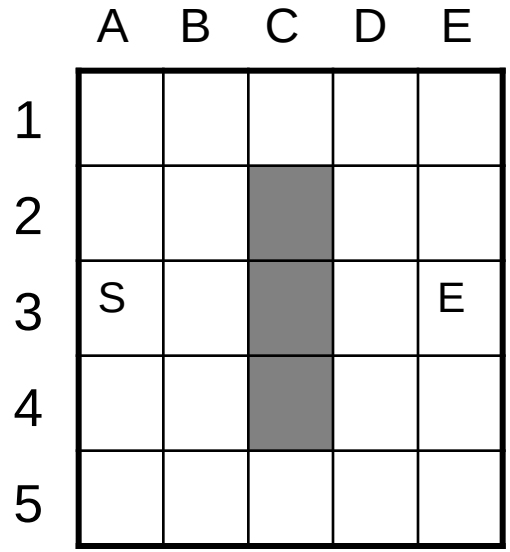
DFS Example 1



DFS Example 2



Grid-Based DFS, BFS



Agenda

- Intro & Admin
- Intro to AI
- Math & Physics Review
- Intro to Pathfinding
- Graphs & Nodes
- Breadth-First Search
- Depth-First Search
- Assignment Overview

Assignment

- Grid-based map, 10 x 10
- Specify map in text file
- Implicit representation of graph by grid
- Different map tiles types
- Bonus: user-specified dimensions