Introduction to Artificial Intelligence

Nathan Sprague
JMU Department of Computer Science

Computer Science Content Teaching Academy
June 26, 2014

Agenda

- Why it is hopeless to give a 90 minute (or one week) overview of AI
- Graph Search (with Exercises)

Problem #1: Definitions

- Even defining AI raises thorny/interesting philosophical questions:
  - How do we define intelligence?
  - What is the connection between intelligence and consciousness? (if any)
  - Is it possible for a machine to be intelligent (or conscious)?

Problem #1: Definitions

- Even defining AI raises thorny/interesting philosophical questions:
  - How do we define intelligence?
  - What is the connection between intelligence and consciousness? (if any)
  - Is it possible for a machine to be intelligent (or conscious)?
- Computing Machinery and Intelligence (Alan Turing, 1950)

Problem #2: AI is a Mess

Problem Domains

Algorithmic Approaches

	Computer Vision	Robotics	Natural Language Understanding	Artificial Creativity	…
Graph Search					
Logic					
Constraint Satisfaction					
Probabilistic/ Statistical methods					
Planning					
Control Theory					
Machine Learning/ Neural Networks					
…					

Problem #3: Modern AI is Mathy

Abstracts from the 2013 International Joint Conference on Artificial Intelligence (IJCAI 13)

Problem #4: AI is Intrinsically Disappointing

- Once a student understands an AI algorithm, it no longer seems ... intelligent.

Problem #4: AI is Intrinsically Disappointing

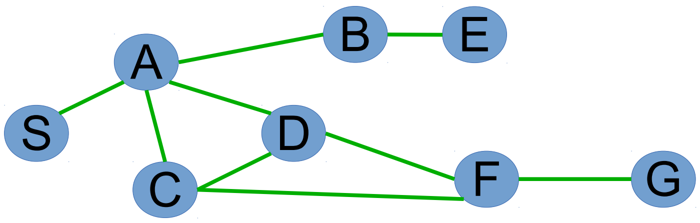
- Once a student understands an AI algorithm, it no longer seems ... intelligent.
- Once a problem is solved it no longer counts as AI
- The problems that are left are *fundamentally* hard

So What to Do?

- In one week you can:
  - Develop a sense for why the problems are hard
  - Dig into some understandable but useful algorithms:
    - Graph search
    - Neural networks

Reminder(?): Graphs

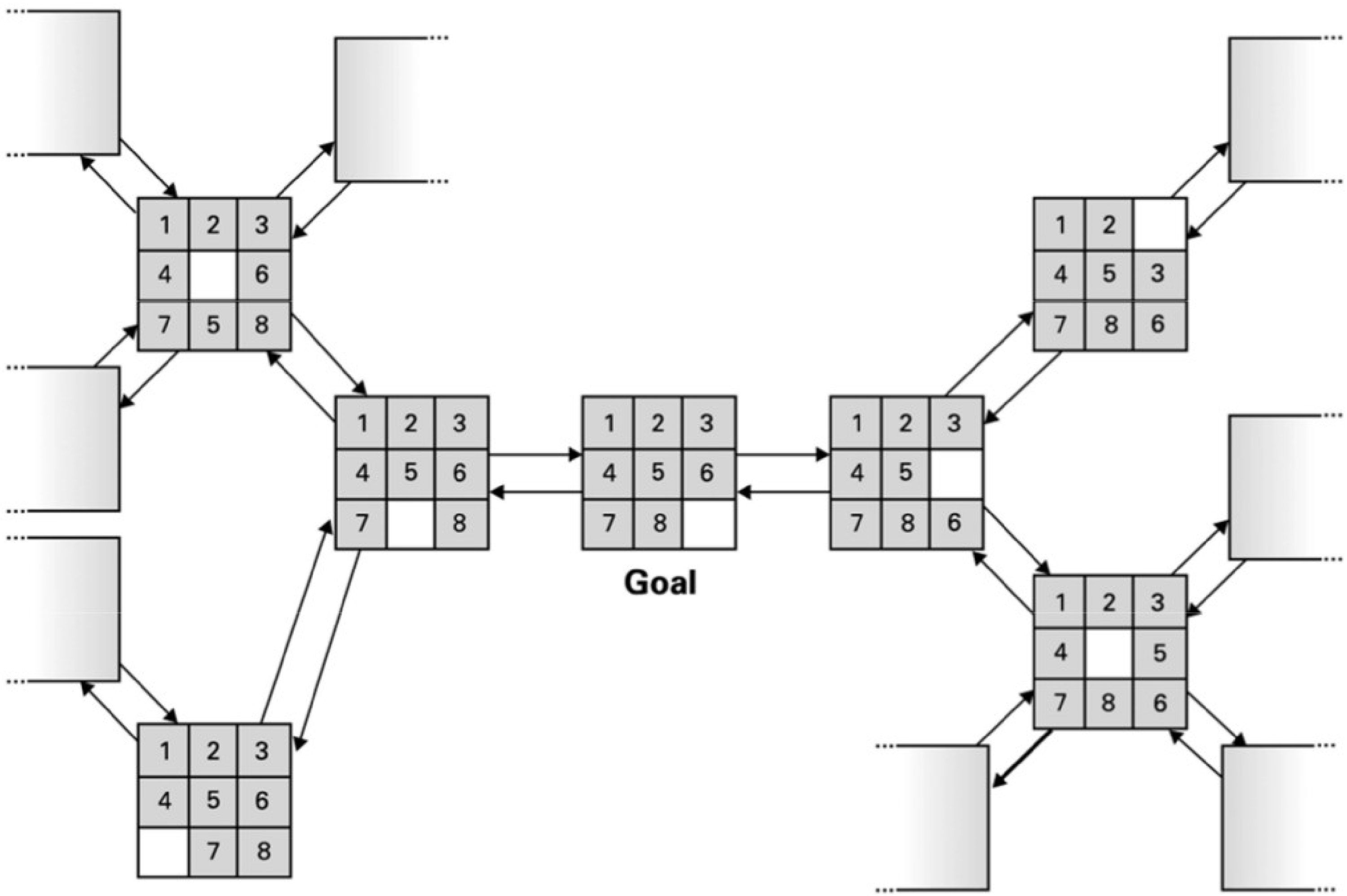
- Very common way of modeling relationships in Computer Science:
  - Set of Nodes/Vertices.
  - Set of Edges connecting nodes.



Modeling Problems

- Many AI problems can be modeled as graphs:
  - Each situation (state) is represented as a node.
    - Specific states are designated as start and goal.
  - Actions are represented as edges.
    - Textbook uses the term productions.

Representing Problems As Graphs: Example

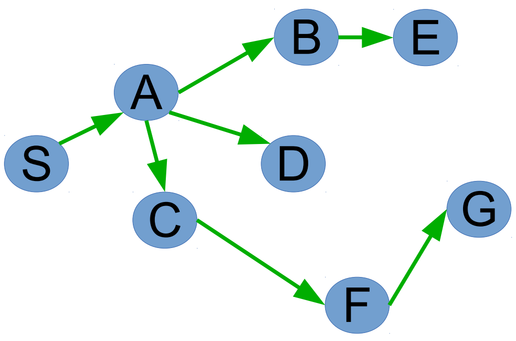


When to Use Graph Search

- Graph search is appropriate when:
  - We know the “rules” of our domain.
  - We always know the current situation (perfect information)
  - The domain is deterministic

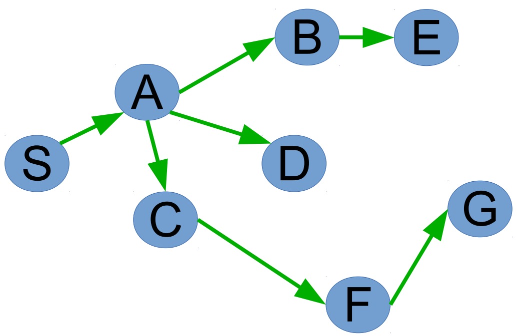
Graph Search Algorithms: Breadth-First Search

```
Procedure BFS(start, goal)
    FRONTIER := An empty Queue.
    FRONTIER.add(start)
    found := False
    while (FRONTIER not empty) and (not found)
        current = FRONTIER.remove()
        if current is the goal then
            found := True
        else
            for each state m accessible from n
                   FRONTIER.add(m)
```



Graph Search Algorithms: Depth-First Search

```
Procedure DFS(start, goal)
    FRONTIER := An empty Stack. //LIFO
    FRONTIER.push(start)
    found := False
    while (FRONTIER not empty) and (not found)
        current = FRONTIER.pop()
        if current is the goal then
            found := True
        else
            for each state m accessible from n
                   FRONTIER.push(m)
```



AISpace – Tools for Learning AI

- Loosely associated with the (freely available) textbook:
  - Artificial Intelligence: Foudations of Computational Agents Poole and Mackworth, 2010.
- Let's take a look at the search tool.

Exercises 1/2

1) DFS Exercise

- Open the AISpace search tool.
- File → Load Sample Problem → Extended Tree Graph
- Open the "Solve" tab.
- Search Options → Depth First Search
- Step through the search using "Fine Step" Button.
  - Can you anticipate which node will be chosen from the frontier?
  - What is the largest number of nodes that end up on the frontier?

2) BFS Exercise

- Search Options → Depth First Search
- Repeat the search.
  - Can you anticipate which node will be chosen from the frontier?
  - What is the largest number of nodes that end up on the frontier?

3) DFS and Loops

- Open the "Create" tab.
- Add an edge from N4 to S.
- Search Options → Depth First Search
- Run the search.
- Search Options → Pruning → Multiple-Path Pruning
- Repeat the search.

4) DFS vs BFS

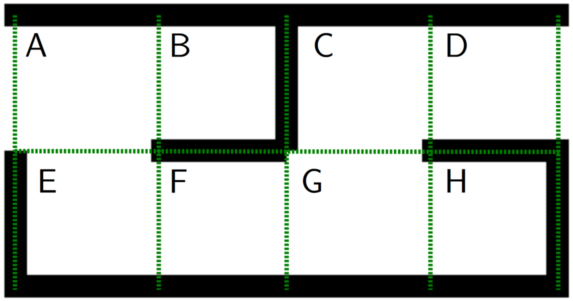
- Remove the edge from N4 to S.
- Add an edge from N20 to N2.
- Search using BFS and DFS.
- Do they find the same path?

Exercises 2/2

DUNGEON ADVENTURE!

1) <u>Dungeon #1</u>

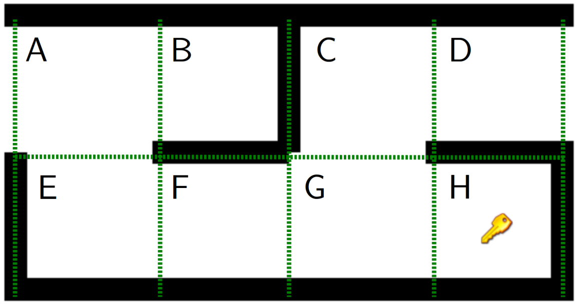
- File → Create New Graph
- Create a graph representation of the following maze. A is the start state. D is the goal.



- Use DFS or BFS to find a path.

2) <u>Dungeon #2</u>

- Modify Dungeon 1 so that:
- There is a key that must be picked up at position H before we can exit the maze.



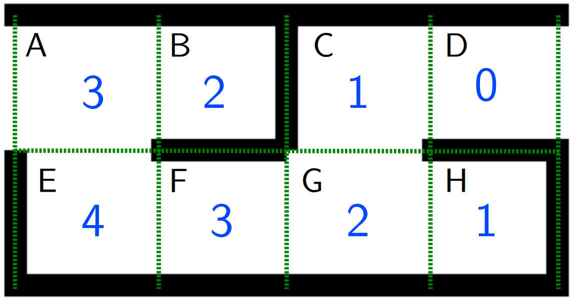
- Once the key is picked up it cannot be put down.
- Use DFS or BFS to find a path.

Problems with BFS/DFS

- Graph search can be VERY expensive
- Chess example:
  - About 80 moves in a game
  - Branching factor of about 30
  - Size of the search tree is something like 30^{80}
- It helps to have an idea of which direction we should go…

Search Heuristics

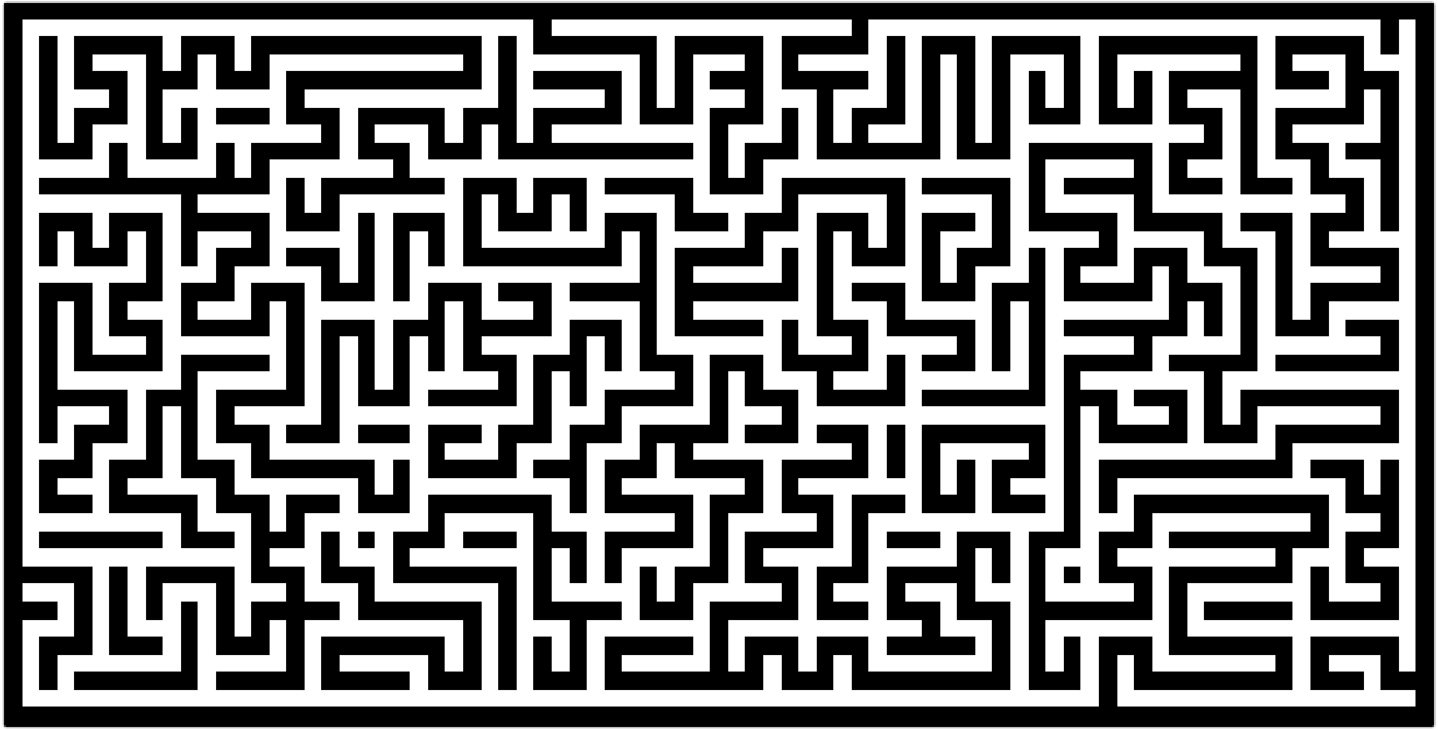
- Heuristic function:
  - An easily calculated estimate of how far a state is from the goal
  - For our maze we could use the # of steps disregarding walls:



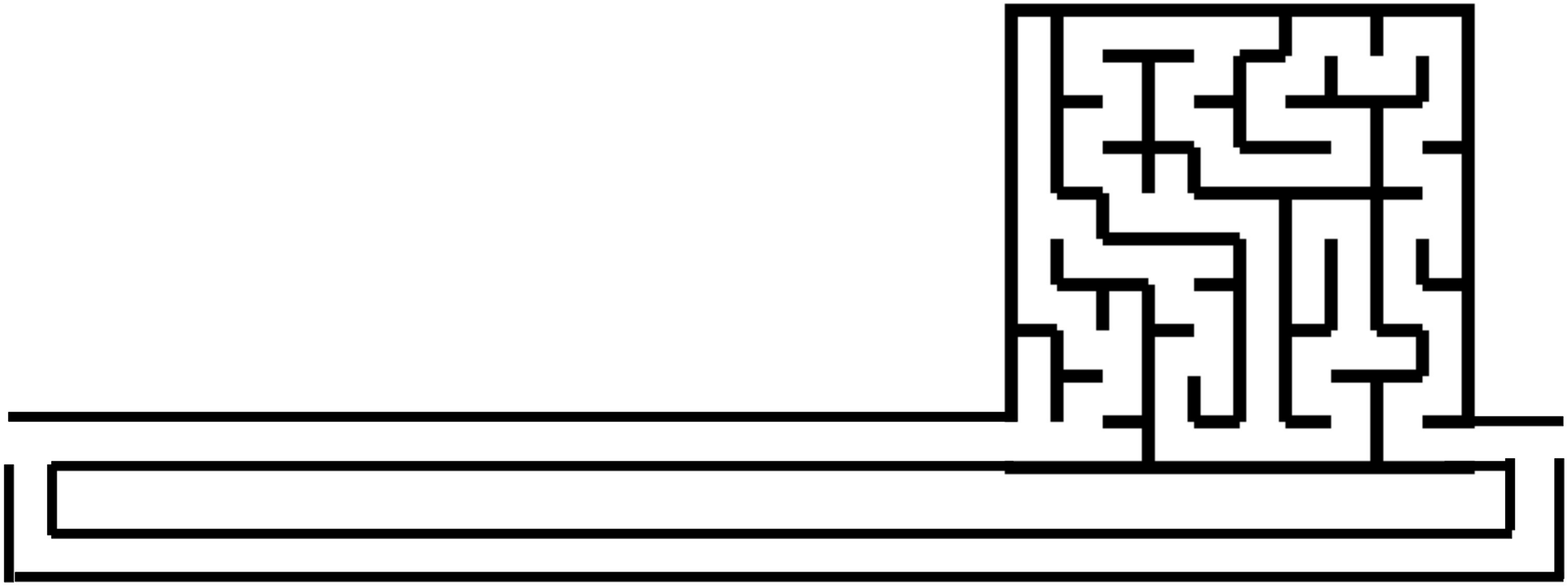
  - Best-First Search: Always choose the frontier node with the smallest heuristic value
  - (Not very helpful for this maze)

Heuristics

- It would probably help for this one:



The Problem with Best-First Search



A* Search

- Probably the most famous AI algorithm.
- Tiny variation on best first search:
  - Instead of choosing the node with the smallest heuristic h(n), choose the node with the smallest sum:

    $f(n) + h(n)$
  - Where f(n) is the observed cost to reach n from the start state.
- A* is guaranteed to find the optimal path.
  - Possibly much more quickly than BFS.

Exercises

1) DFS Exercise

- What is the largest number of nodes that end up on the frontier?

  4

- In general $b * d$ where b is the branching factor of the graph and d is the depth of the search.

2) BFS Exercise

- What is the largest number of nodes that end up on the frontier?

  6

- In general b^d.

3) DFS and Loops

– Add an edge from N4 to S.

– DFS is stuck in an infinite loop!

– Search Options $\rightarrow$ Pruning $\rightarrow$ Multiple-Path Pruning

– Nodes are only added to the frontier if

- They are not already on the frontier

  and

- Have never been selected.

4) DFS vs BFS

– Do they find the same path?

– No! BFS is guaranteed to find the path with the fewest steps. DFS is not.