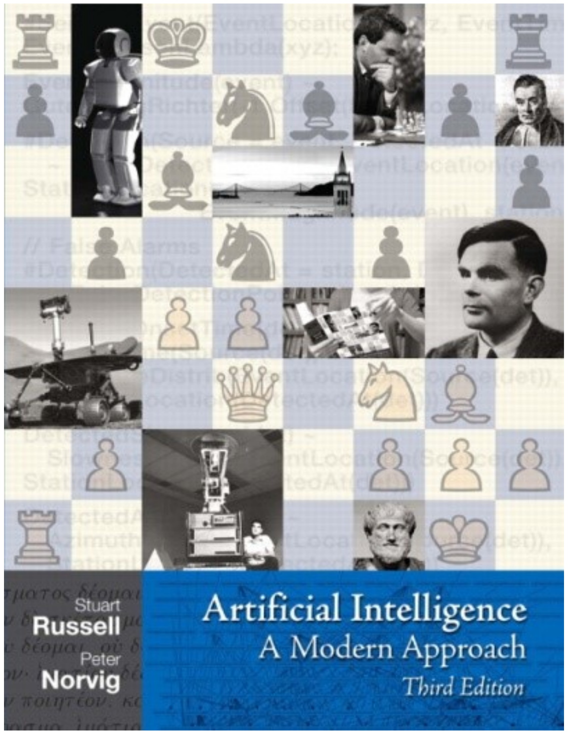
Artificial Intelligence

Course Information

- Prerequisites:
  - Programming
  - Data Structures and Algorithms
  - Probability, Linear Algebra (Optional)
- Grading:
  - Continuous evaluation through assignments and/or quizzes
  - Mid-term/End-term open-book/take-away exams?

Textbook

- Russell & Norvig, AI: A Modern Approach, 3rd Ed.



What is AI?

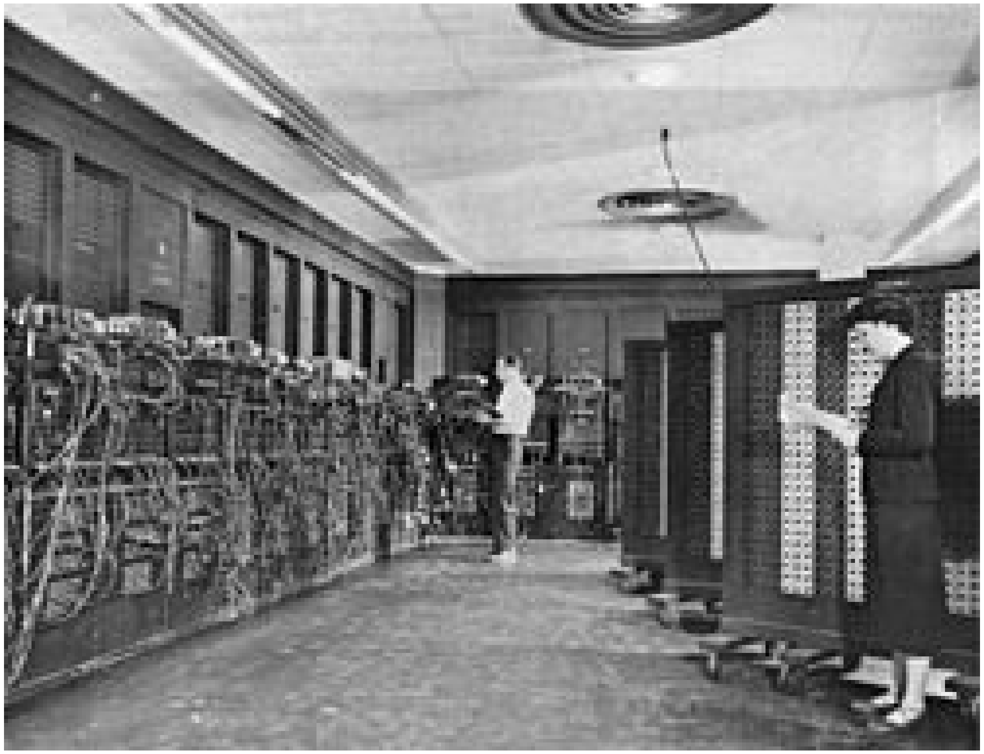
- Study of how to make computers do things which at the moment, people do better
- Earlier work focused on game playing and theorem proving
  - Game playing and theorem proving share the property that the people who do well in it, display "Intelligence"
  - e.g. Checkers-playing program, Logic theorist

What is AI?

- Another area was “commonsense reasoning”
  - Reasoning about physical objects, their relationships, actions and consequences
  - e.g. General Problem Solver (by Newell, Shaw, and Simon) applied to the problem of symbolic manipulations of logical expressions
  - Only simple tasks were selected
- Progress in AI research and computation led to solving more complex problems related to perception (vision, speech), natural language processing, medical diagnosis etc.

AI History

- ENIAC, 1946



AI History

- 1950, Turing asked the question:

  "Can machines think?"

AI History

- 1956, A new field was born:

  "We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire."

  – Dartmouth AI Project Proposal; J. McCarthy et al.; Aug. 31, 1955

AI History

- 1964: Eliza –the chatbotpsychotherapist

  -Reasoning about physical objects, their relationships, actions and consequences

- 1966: Shakey–general purpose mobile robot

AI History

- 1964: Eliza –the chatbot psychotherapist
  - First attempt to interact with human
  - Used “Pattern Matching ” and “Substitution” methodology
  - no built in framework for contextualizing events

AI History

- 1966: Shakey– general purpose mobile robot
  - First mobile, intelligent robot
  - Used the LISP programming language
  - Used three-tier software architecture:
    - ➢ Low level actions/reflexes:e.g. Moving forward or moving camera eyes
    - ➢ Intermediate level actions: Pushing an object across a surface
    - ➢ High level actions: Moving from one room to another while dealing with unexpected obstacles
  - declared an "IEEE Milestone"
  - combined research in robotics, computer vision, and natural language processing
  - results of the project include the A* search algorithm, the Hough transform, and the visibility graph

AI History

- First AI Winter (1974-1980):
  - Failure of Machine Translation
  - Poor speech understanding
  - Negative results in neural nets
- Second AI Winter (1987-1993):
  - Decline of LISP
  - Decline of specialized hardware for expert systems

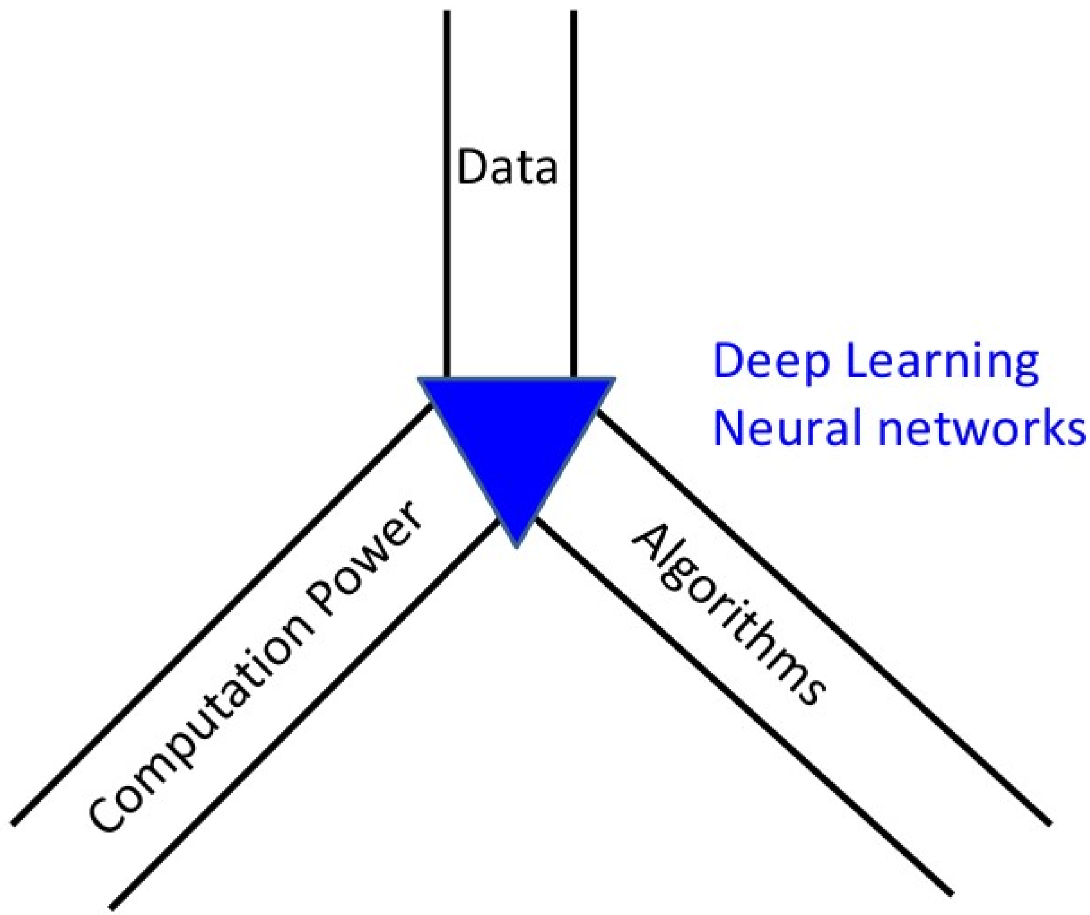
AI History

- Statistical approaches (1990-):
  - Focus on uncertainty
  - e.g. SVM, KNN, regression, clustering etc.
  - Based on hand-crafted features

Current State of AI

- Increase in AI funding- How?
  - Success in:
    - ➢ Image recognition (ImageNet training database-2012)
    - ➢ Language translation (Google Translate)
    - ➢ Game playing system such as AlphZero(chess champion), AlphaGo(go champion) and Watson (Jeopardy champion)
    - ➢ Due to advent of deep learning

What Changed?



What Changed?

- 1960’s : Realization that computers could simulate 1-layer neural networks led to hype-cycle
- 1969: Evident that only a limited set of problems could be optimally solved by 1-layer neural networks
- 1985: Neural networks could be used to solve optimization problems (due to famous paper by Hopfield and Tank)
- 2000’s: Advent of GPUs and custom VLSI chips increased the computation power, which led to shift from 1-layer neural networks to 2-layer neural network research

What Changed?

- 2010's: Advent of Big Data due to:
  - Availability of cheap labour e.g. Amazon Mechanical Turk
  - Social media leading to availability of text data
  - Significant increment in the voice based queries due to availability of voice-based-intelligent assistants such as SIRI, Alexa, Facebook's M, Google home etc
  - Big image database such as ImageNet leading to significant progress in Computer Vision, Medical Imaging research

Artificial Intelligence (Agent View)

- An agent is anything that can be viewed as “perceiving” its environment through “sensors” and acting upon that environment through “actuators”

Artificial Intelligence (Agent View)

- An agent is anything that can be viewed as “perceiving” its environment through “sensors” and acting upon that environment through “actuators”
- Human agent:

Artificial Intelligence (Agent View)

- An agent is anything that can be viewed as “perceiving” its environment through “sensors” and acting upon that environment through “actuators”
- Human agent:
  - Sensors: eyes, ears, and other organs

Artificial Intelligence (Agent View)

- An agent is anything that can be viewed as “perceiving” its environment through “sensors” and acting upon that environment through “actuators”
- Human agent:
  - Sensors: eyes, ears, and other organs
  - Actuators: hands, legs, mouth, and other body parts

Artificial Intelligence (Agent View)

- An agent is anything that can be viewed as "perceiving" its environment through "sensors" and acting upon that environment through "actuators"
- Human agent:
  - Sensors: eyes, ears, and other organs
  - Actuators: hands, legs, mouth, and other body parts
- Robotic agent:

Artificial Intelligence (Agent View)

- An agent is anything that can be viewed as “perceiving” its environment through “sensors” and acting upon that environment through “actuators”
- Human agent:
  - Sensors: eyes, ears, and other organs
  - Actuators: hands, legs, mouth, and other body parts
- Robotic agent:
  - Sensors: cameras and laser range finders

Artificial Intelligence (Agent View)

- An agent is anything that can be viewed as "perceiving" its environment through "sensors" and acting upon that environment through "actuators"
- Human agent:
  - Sensors: eyes, ears, and other organs
  - Actuators: hands, legs, mouth, and other body parts
- Robotic agent:
  - Sensors: cameras and laser range finders
  - Actuators: various motors, mechanical arms

Artificial Intelligence (Agent View)

- Performance measure: An objective criterion for success of an agent's behavior

Artificial Intelligence (Agent View)

- Performance measure: An objective criterion for success of an agent's behavior
- E.g., performance measure of a vacuum-cleaner agent could be amount of dirt cleaned up, amount of time taken, amount of electricity consumed, amount of noise generated, etc.

Artificial Intelligence (Algorithmic View)

- A large number of problems are NP hard

Artificial Intelligence (Algorithmic View)

- A large number of problems are NP hard
  - NLP: Sentence generation, Machine Translation

Artificial Intelligence (Algorithmic View)

- A large number of problems are NP hard
  - NLP: Sentence generation, Machine Translation
  - Computer Vision (CV): Object recognition/segmentation

Artificial Intelligence (Algorithmic View)

- A large number of problems are NP hard
  - NLP: Sentence generation, Machine Translation
  - Computer Vision (CV): Object recognition/segmentation
- AI develops a set of tools, heuristics, ...

Artificial Intelligence (Algorithmic View)

- A large number of problems are NP hard
  - NLP: Sentence generation, Machine Translation
  - Computer Vision (CV): Object recognition/segmentation
- AI develops a set of tools, heuristics, ...
  - to solve such problems in practice

Artificial Intelligence (Algorithmic View)

- A large number of problems are NP hard
  - NLP: Sentence generation, Machine Translation
  - Computer Vision (CV): Object recognition/segmentation
- AI develops a set of tools, heuristics, ...
  - to solve such problems in practice
- Will look at Planning/Search, Game Playing and heuristics used to solve some of these problems

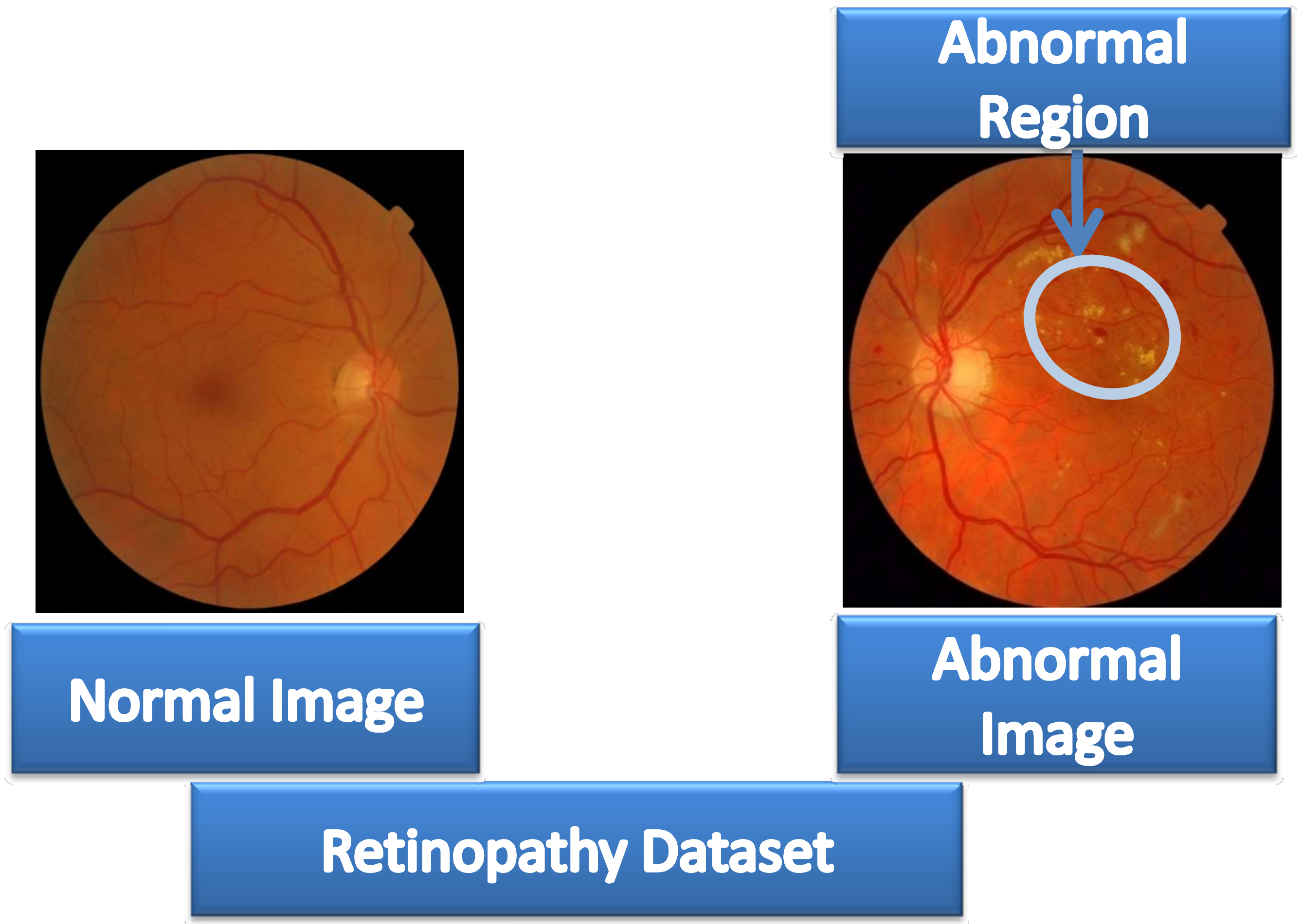
(Rational) Agent

- Input:
  - Set of states (e.g. through Graph)
  - Start state
  - Goal test
  - Successor function
  - Cost associated with every action
- Output:
  - Path: Start state => Success (state satisfying the goal test) (e.g. condition may require the shortest path)
  - or Start state => failure

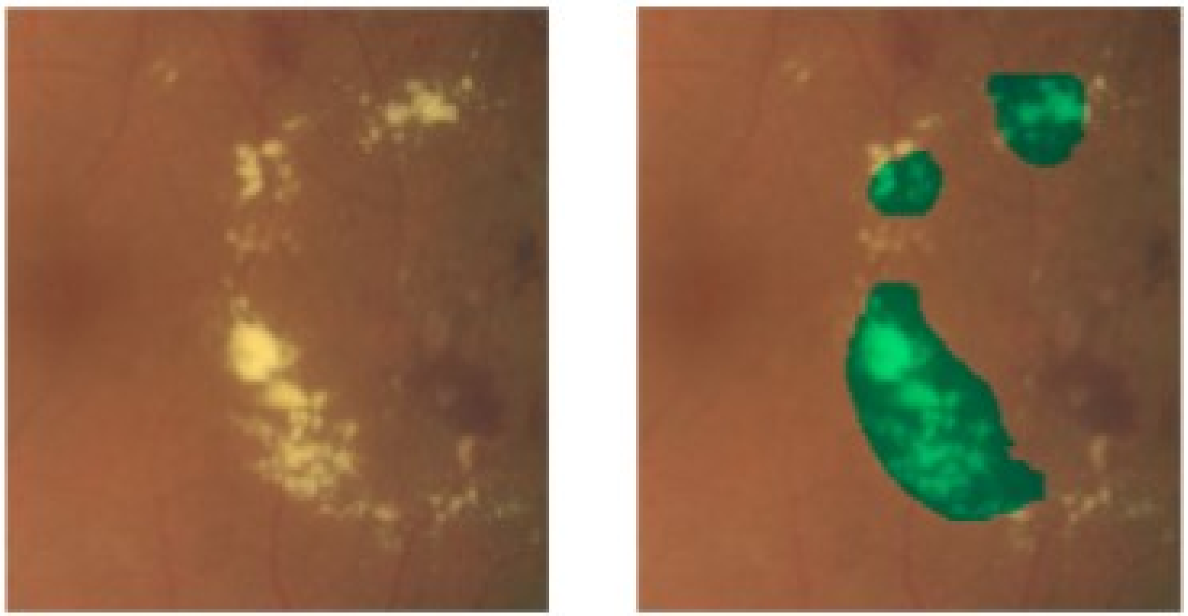
Why is Searching Interesting?

- Many problems can be formulated as search problems
  - Path planning (Jaipur to Mumbai)
  - Games (e.g. 8-puzzle problem)
  - Natural Language Processing (e.g. think of what happens when you write a mail through your gmail account or writing queries on google search engine)
  - Robotics (e.g. Shakey moving from one room to another using algorithm which uses visibility graph)
  - Machine Learning (e.g. Abnormality detection and segmentation, curve fitting)

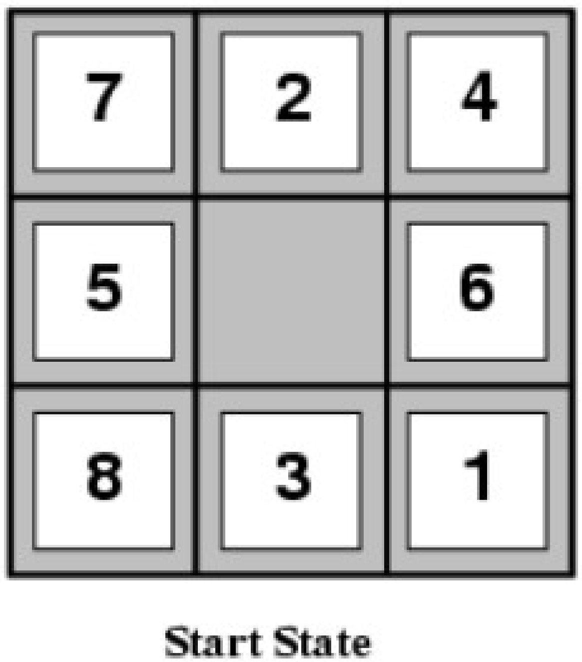
Abnormality Detection/Segmentation in Medical Images

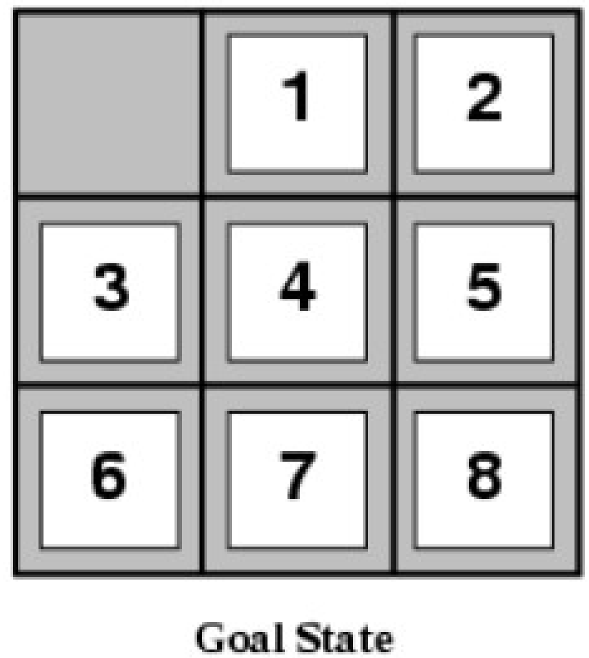


Abnormality Detection/Segmentation in Medical Images



Example: 8-puzzle

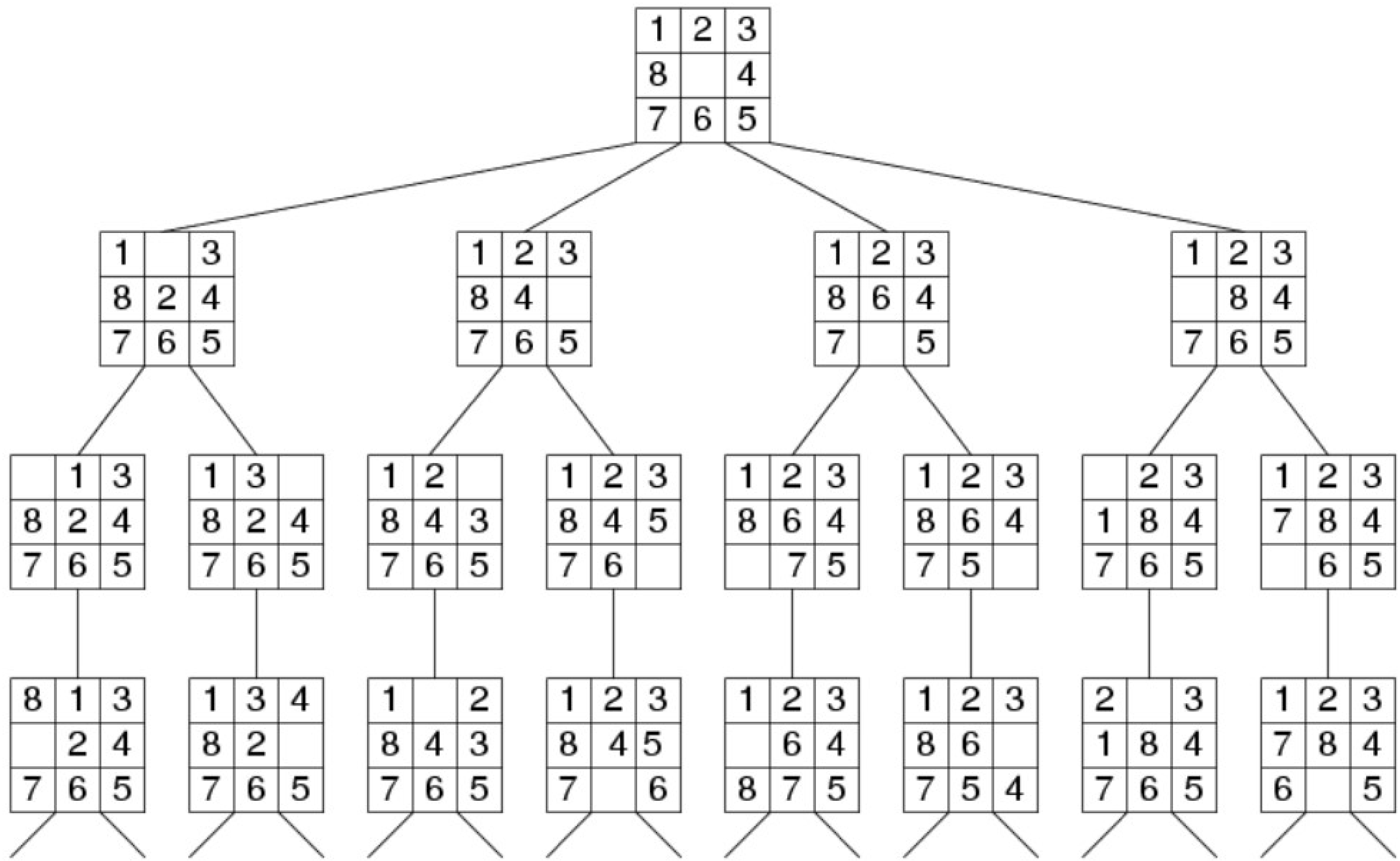




Example: 8-puzzle

- states?
  - locations of tiles
- actions?
  - move blank left, right, up, down
- goal test?
  - goal state (given)
- path cost?
  - 1 per move
- [Note: optimal solution of n-Puzzle family is NP-hard] why?

Example: 8-puzzle Search Tree (fragment)



Example: 8-puzzle Search Tree (fragment)

- A state is a situation that an agent can find itself in
- We distinguish two types of states:
  - world states
    - The actual concrete situations in the real world
  - representational states
    - abstract descriptions of the real world that are used by the agent in deliberating about what to do

Example: 8-puzzle Search Tree (fragment)

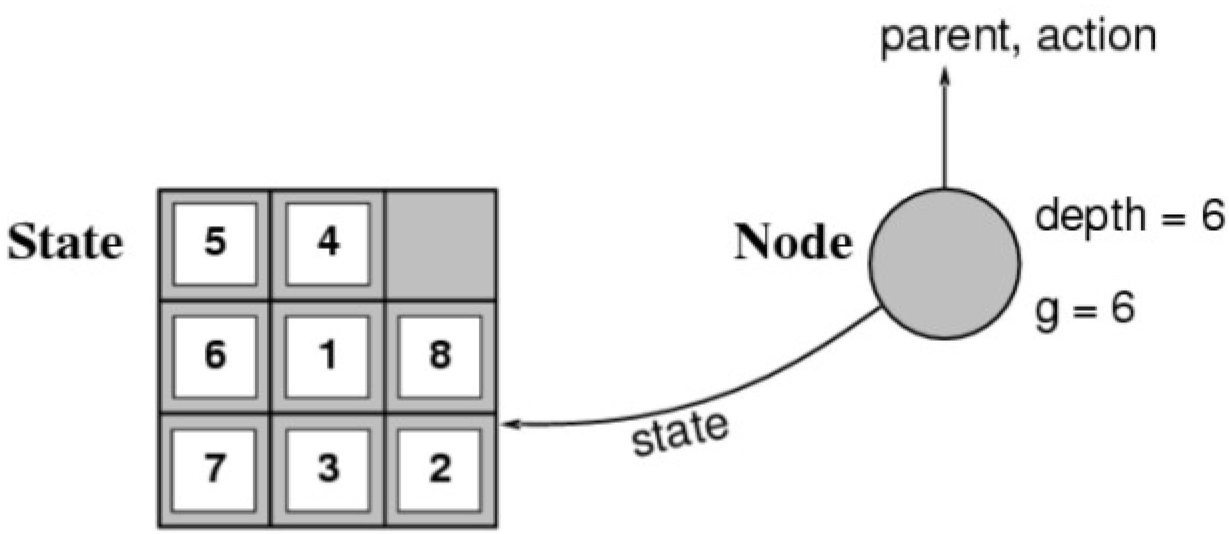
- A state space is a graph whose nodes are the set of all states, and whose links are actions that transform one state into another
- A search tree is a tree (a graph with no undirected loops) in which the root node is the start state and the set of children for each node consists of the states reachable by taking any action

Example: 8-puzzle Search Tree (fragment)

- A search node is a node in the search tree
- A search node is a data structure constituting part of a search tree that includes state, parent node, action, path cost g(x), depth
- A goal is a state that the agent is trying to reach
- An action is something that the agent can choose to do
- A successor function described the agent's options: given a state, it returns a set of (action, state) pairs, where each state is the state reachable by taking the action
- The branching factor in a search tree is the number of actions available to the agent

Example: 8-puzzle Search Tree (fragment)

- The Expand function creates new nodes, filling in the various fields and using the SuccessorFn of the problem to create the corresponding states.



Search strategies

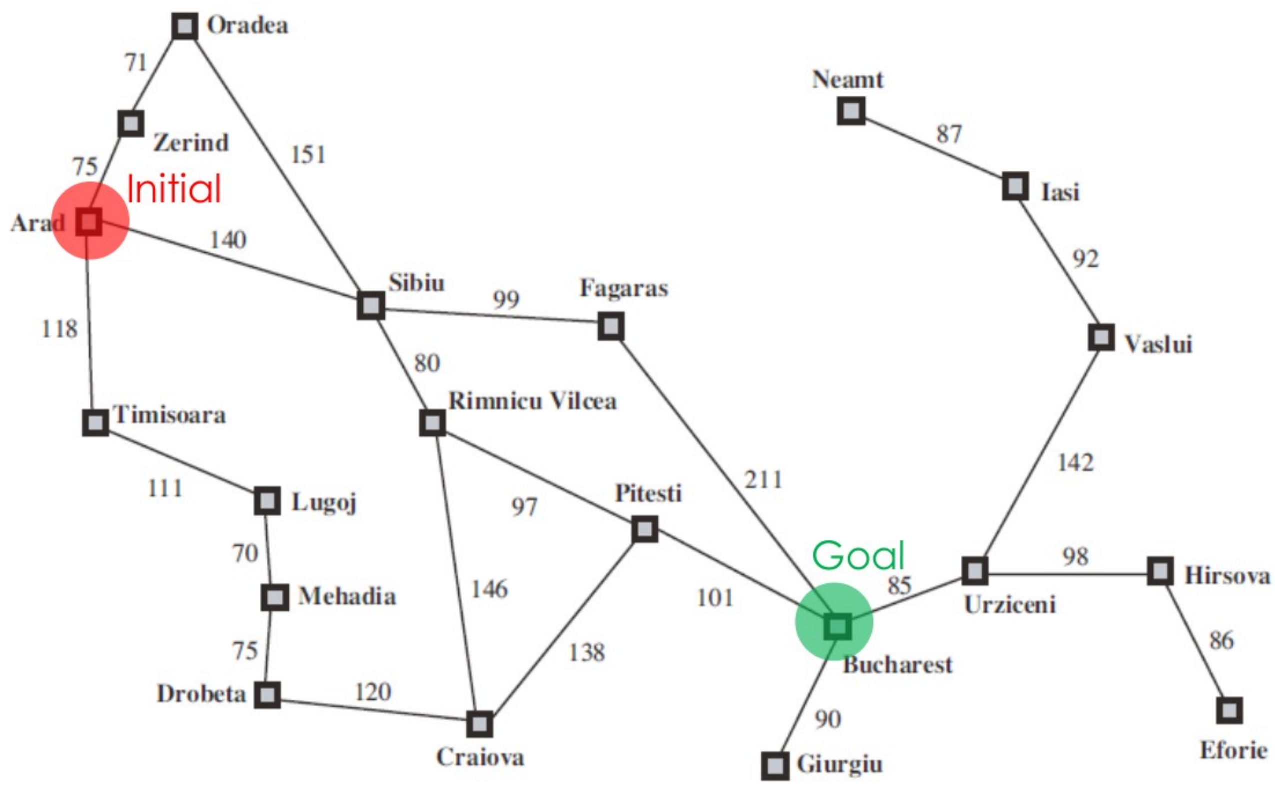
- A search strategy is defined by picking the order of node expansion
- Strategies are evaluated along the following dimensions:
  - completeness: does it always find a solution if one exists?
  - time complexity: number of nodes generated
  - space complexity: maximum number of nodes in memory
  - optimality: does it always find a least-cost solution?
  - systematicity: does it visit each state at most once?
- Time and space complexity are measured in terms of
  - b: maximum branching factor of the search tree
  - d: depth of the least-cost solution
  - m: maximum depth of the state space (may be ∞)

Uninformed Search Strategies

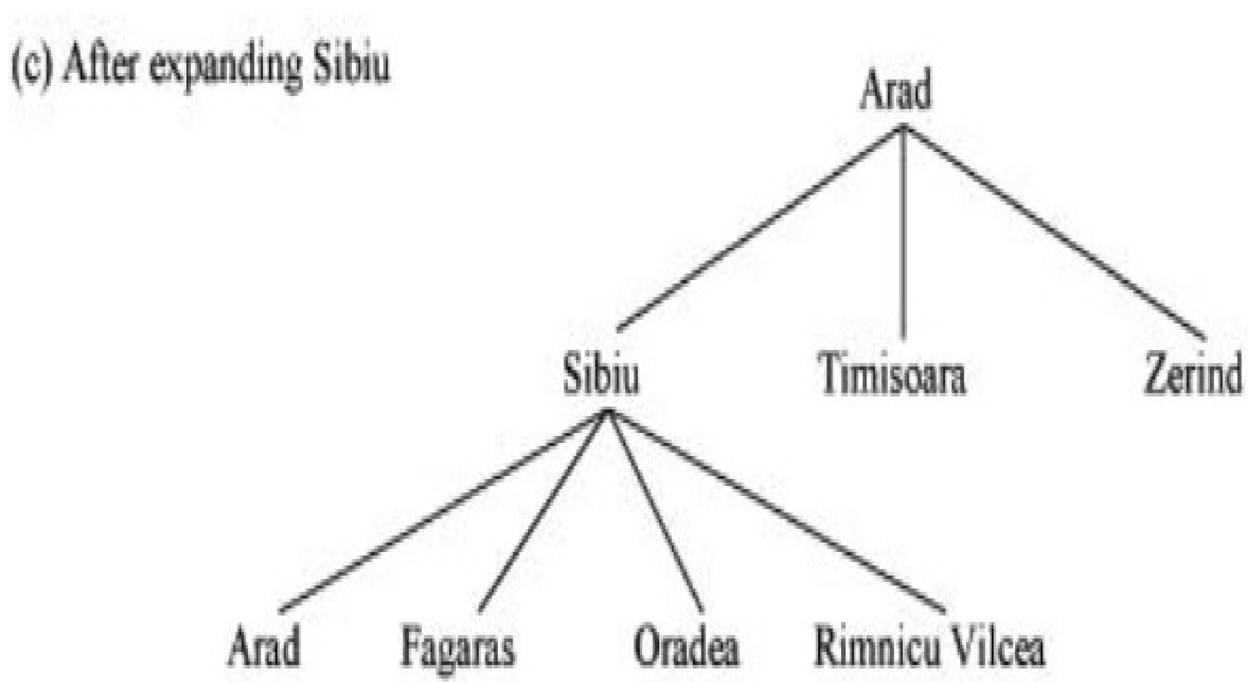
- Uninformed search strategies use only the information available in the problem definition
  - Breadth-first search
  - Depth-first search
  - Depth-limited search
  - Iterative deepening search

Repeated States

- Generated in the search tree due to redundant paths. e.g.

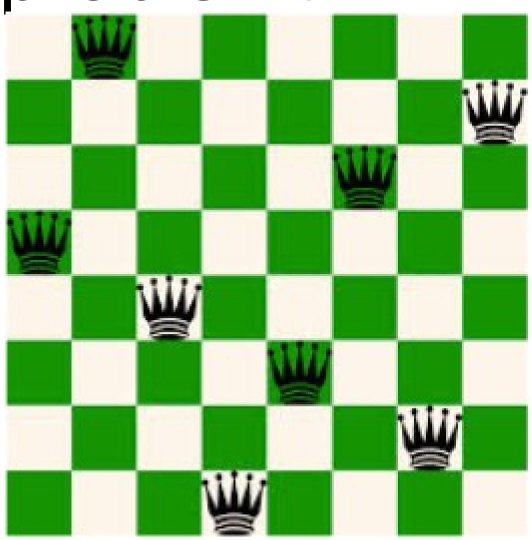


Repeated States



Repeated States

- What you can say about the 8-puzzle problem?
- What you can say about the 8-Queens problem?



- State: Configuration of the Queens
- Initial State: Empty board
- Goal Test: Configuration with 8 queens on the board with none attacking another
- Path Cost: Time taken to solve?

Search Trees

Function TREE-SEARCH(problem, strategy) returns a solution or a failure

initialize the search tree using the initial state of the problem

loop do

If there are no candidates for expansion then return failure

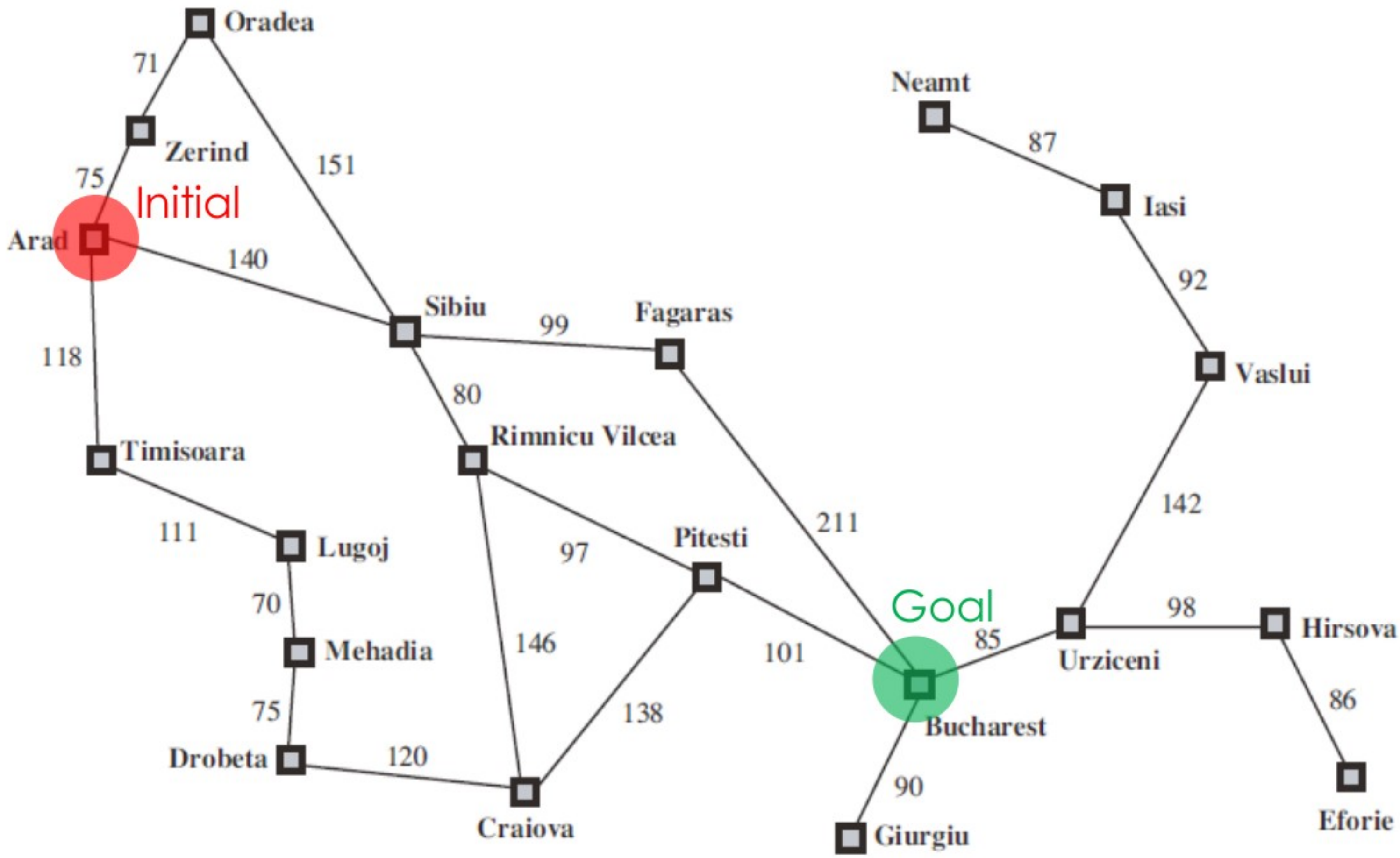
choose a leaf node for expansion according to the strategy

If the node contains a goal state then return the corresponding solution

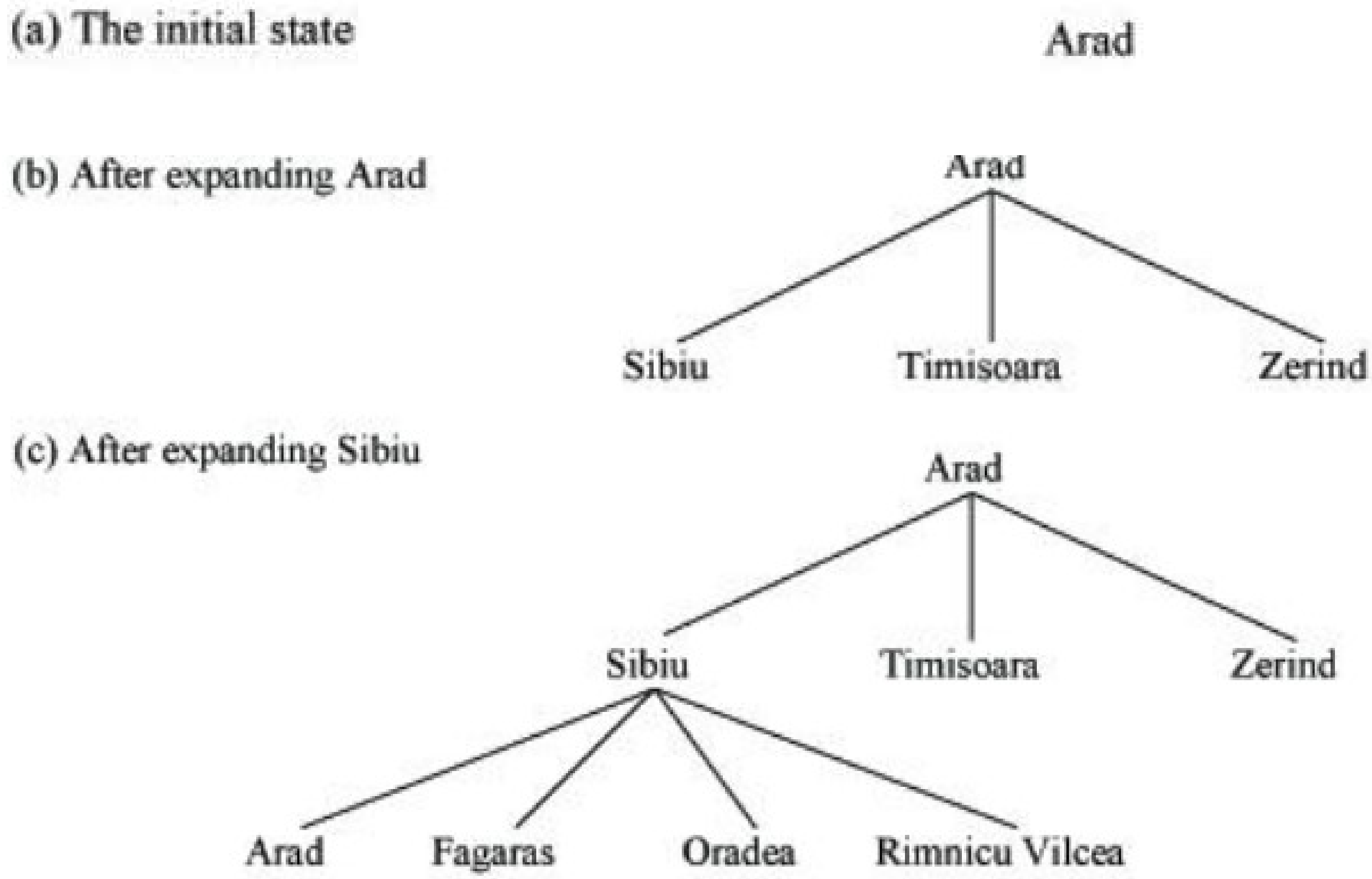
Else expand the node and add the resulting nodes to the tree

end

Example


Oradea
71
Neamt
Zerind
151
87
75
Initial
Iasi
Arad
140
92
Sibiu
99
Fagaras
118
Vaslui
80
Rimnicu Vilcea
Timisoara
142
111
211
Lugoj
97
Pitesti
70
Goal
98
Hirsova
85
Mehadia
146
101
Urziceni
86
75
138
Bucharest
Drobeta
120
90
Craiova
Giurgiu
Eforie

Example


(a) The initial state
Arad
(b) After expanding Arad
Arad
Sibiu
Timisoara
Zerind
(c) After expanding Sibiu
Arad
Sibiu
Timisoara
Zerind
Arad
Fagaras
Oradea
Rimnicu Vilcea

Graph Search

Function GRAPH-SEARCH(problem) returns a solution or a failure

initialize the frontier using the initial state of the problem

initialize the explored set to be empty

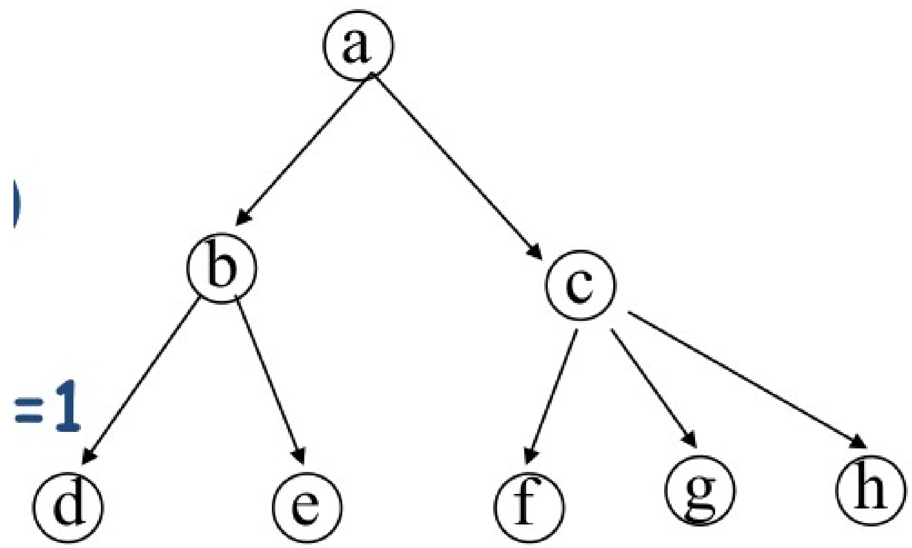
loop do

- If the frontier is empty then return failure
- choose a leaf node and remove it from the frontier
- If the leaf node contains a goal state then return the corresponding solution
- Add the node to the explored set
- Expand the chosen node, adding the resulting nodes to the frontier only if not in the frontier or explored set

end

Breadth First Search: shortest/Shallowest first

- Expand shallowest unexpended node
- Implementation: FIFO Queue; successors at the end of the queue
- Evaluation
  - Complete? Yes (b is finite)
  - Time Complexity? O(b^d)
  - Space Complexity? O(b^d)
  - Optimal? Yes, if stepcost=1



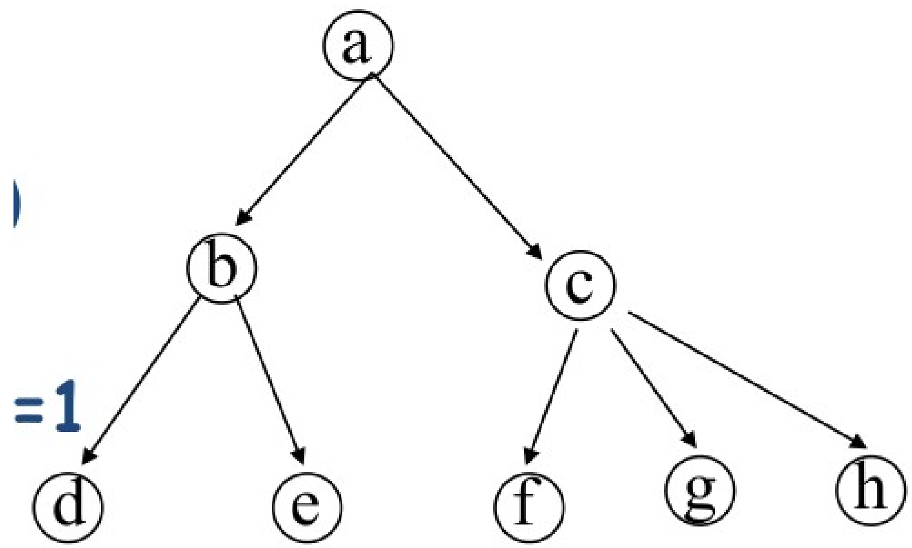
Issues with Breadth First Search

Depth	Nodes	Time	Memory
2	110	.11 milliseconds	107 kilobytes
4	11,110	11 milliseconds	10.6 megabytes
6	10^6	1.1 seconds	1 gigabyte
8	10^8	2 minutes	103 gigabytes
10	10^{10}	3 hours	10 terabytes
12	10^{12}	13 days	1 petabyte
14	10^{14}	3.5 years	99 petabytes
16	10^{16}	350 years	10 exabytes

Figure 3.13 Time and memory requirements for breadth-first search. The numbers shown assume branching factor $b = 10$; 1 million nodes/second; 1000 bytes/node.

Depth First Search: Deepest first

- Expand Deepest unexpended node
- Implementation: LIFO Queue; successors at the front of the queue
- Evaluation
  - Complete? No
  - Time Complexity? O(b^m)
  - Space Complexity? O(bm)
  - Optimal? No

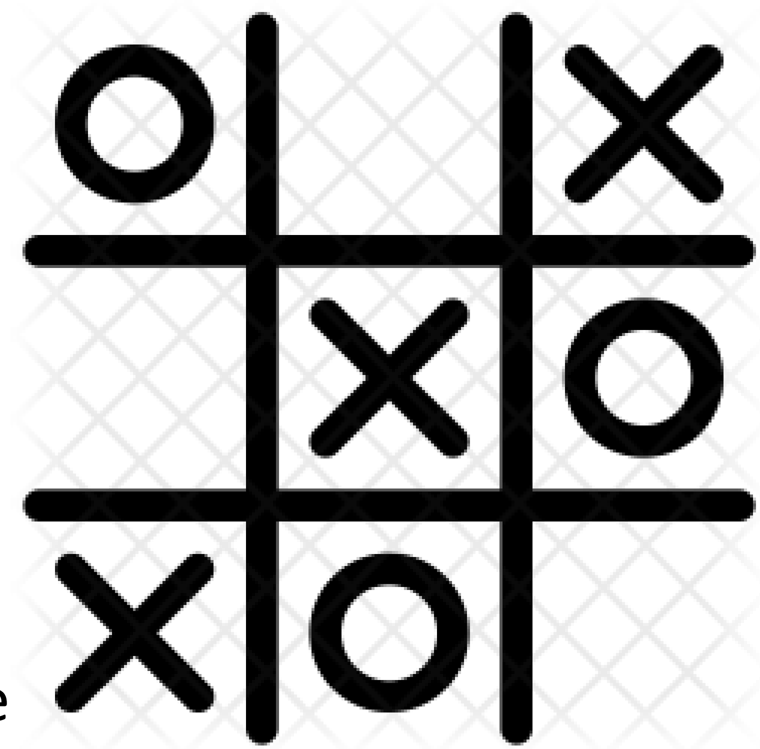


Depth Limited Search

- Depth-first search with depth limit l
- Implementation: nodes at depth l have no successors
- Only finite space to be explored
- Completeness: Yes/No?
- Optimality: Yes/No?

Tic-Tac-Toe

- The first player is X and second is 0
- Object of game: Get three of a player's symbol in horizontal, vertical or diagonal row
- X always goes first
- Players alternate placing Xs and 0s on the game board
- Game ends when a player has three in a row (a wins) or all nine squares are filled (a draw)



Exercises

- In the game of chess which strategy you would like to prefer to find the winning state?
- In Tic-Tac-Toe, define the states, initial-state, actions, goal test and path cost.
- Whether the tree associated with the Tic-Tac-Toe will be a search tree?
- Tic-Tac-Toe, which strategy should be adopted to avoid loosing the game?
- Whether the search tree of Tic-Tac-Toe will be having the redundancy in terms of paths or nodes? Justify your answer.

Exercises

- Go to the following website:

  https://www.pandorabots.com/mitsuku/

  Observe the Chabot behavior and find the strengths and deviations while having chatting with it.

Next Class

- Iterative Deepening dept-first search, Bidirectional search
- Uniform cost search
- Informed Search Strategies A* search

References

- “Artificial Intelligence: A modern aproach” by Russell and Norvig
- “Artificial Intelligence. Structures and Strategies for Complex Problem Solving” by George F. Luger. Sixth Edition-Addison Wesley (2009)
- UCB CS-188 course
- Professor Mausam IIT Delhi slides