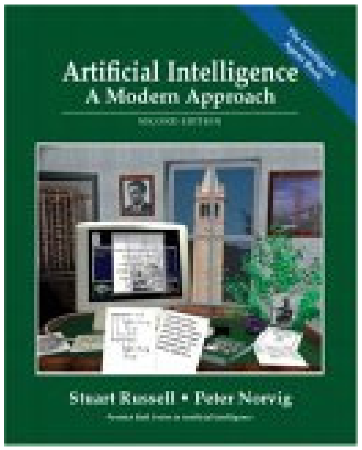
ذكاء اصطناعي
Artificial Intelligence (AI)
Comp 408

Book:



Artificial Intelligence :

A Modern Approach

Second Edition by Stuart J. Russell & Peter Norvig

(2003, Prentice Hall)

- Lectures, Hws, …etc will be available on
- http://scholar.cu.edu.eg/?q=areegsaid/node/25645

AI

G. F. Luger (2005)

- AI is the branch of computer science concerned with the automation of intelligent behavior.

The sources of studying AI

- Philosophy: Logic, methods of reasoning
- Mathematics: Formal representation and proof algorithms, (un)decidability, probability
- Economics: Decision theory
- Neuoroscience: physical state for mental activity
- Psychology: perception, experimental techniques
- Linguistics, psycholinguistics
- Control System: design systems that maximize an objective function

Tentative syllabus for an AI course

Search (chapters 3,4,5,6)

Logic (chapters 7,8,9)

Natural Language Processing (chapter 22,23)

Remark: Comp 408(a) will not necessarily finish the above

Grades

60 % Final

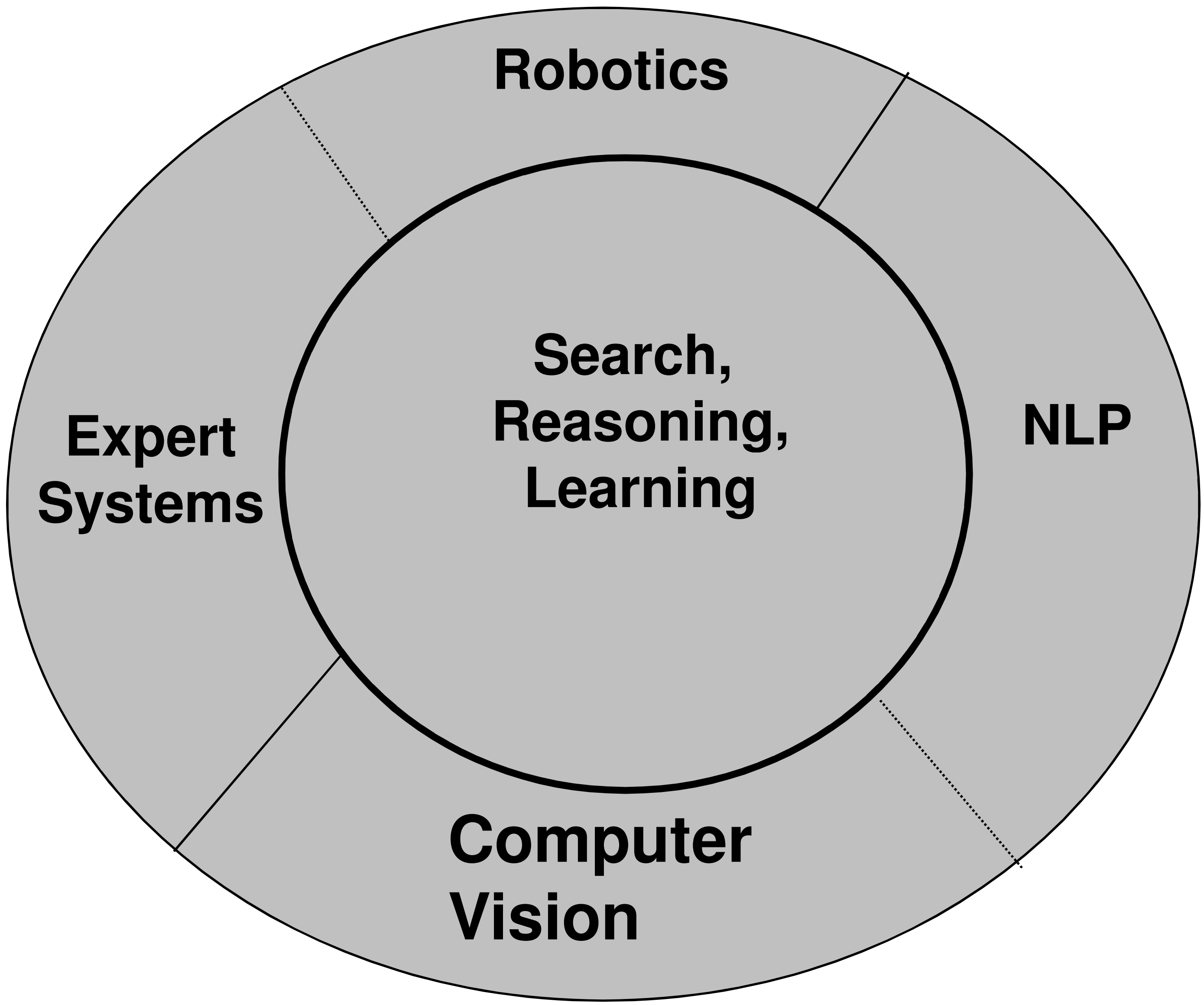
10 % Midterm

10 % Quiz

20 % Lab Assignments/ Quiz/ project

Hw will be assigned frequently to help you study. But no grades on it.

Disciplines which form the core of AI inside
Fields which draw from these discipline ouside.



Human vs Machine Intelligence

- Intelligence is not a unique and unshared capability of human. It is more an open collection of attributes than it is a single well-defined entity
- Humans embody many aspects of intelligence while animals typically embody a smaller number of intelligent characteristics, and usually at a much lower level
- The advent of digital computers made possible credible attempts to fulfill the AI dreams
- Computer based intelligence must be specialized to very *restricted* domains to be comparable to human performance

Human vs Machine Intelligence

- The two hemispheres of the human brain deal with problems in two distinct paradigms:
  - sequential (or logical) approach that considers only a small portion of the available data at a time
  - parallel processing looks at data on a global basis

Many tasks which we might reasonably think require **intelligence** are performed by computers **without even thinking** → Complex Arithmetic

Other tasks that people do without thinking are extremely difficult to automate → Recognizing a Face

What is AI

For Defining AI, Books go in four directions:

- Think Like a human
- Act Like a human
- Think rationally
- Act rationally

The textbook adopts "acting rationally"

Thinking humanly

- 1960s "cognitive revolution"
- How the computer performs tasks does matter
- The reasoning steps are important
- Requires scientific theories of the activities of the brain
- Need ability to manipulate symbols
- How to validate? Requires

  1) Predicting and testing behavior of human subjects (top-down)--- Cognitive Science or

  2) Direct identification from neurological data (bottom-up) cognitive neuroscience

Both approaches are now distinct from AI Approaches

Thinking rationally

Aristotle: what are correct arguments processes?

- Logic and reasoning
- Requires reasoning structures
- Not easy to represent informal information
- Approaches: Machine Learning, NN, Reasoning
- Problems:
  - Not all things can be formally represented in logic notation, particularly if there is any uncertainty
  - It is usually the case that even small scale problems can exhaust the computational power of any computer unless heuristics are used

Acting humanly

- Turing (1950) "Computing machinery and intelligence‟
- "Can machines think?" → "Can machines behave intelligently?"
- Take a task that people normally do : playing chess, diagnosis a disease, navigating a building … etc

  then build a computer system does is automatically

Acting humanly

To pass the Turing Test, the computer would need:

1. Natural Language Processing → Communication
2. Knowledge Representation →store info before and during interrogation
3. Automated Reasoning→ answer questions and draw new conclusions
4. Machine learning→ adapt to new circumstances

Acting Rationally: rational agent

- Rational behavior: doing the right / best thing
- The right thing: maximize goal achievement, given the available information
- Does not necessarily involve thinking
- Act rationally **=** reason logically to the conclusion and act on that conclusion

The Rational Agent Approach

- An agent is something that perceives and acts
- A rational agent is one that acts so as to achieve the best outcome

AI Languages

- Programming languages best suited to AI tasks are Lisp (1960) and Prolog (1972).
  - PROgramming in LOGic
  - Emphasis on *what* rather than *how*
- There also have been specialized knowledge representation systems and languages, used to develop knowledge bases and knowledge-based systems. This includes expert systems, in which probability and beliefs play an important role.

Prolog's strong and weak points

- Assists thinking in terms of *objects* and *entities*
- *Straight forward translation of logic statements*
- Not good for *number manipulation*
- Useful applications of Prolog in
  - *Expert Systems* (Knowledge Representation and Inferencing)
  - *Natural Language Processing*
  - *Relational Databases*

The role of search in AI

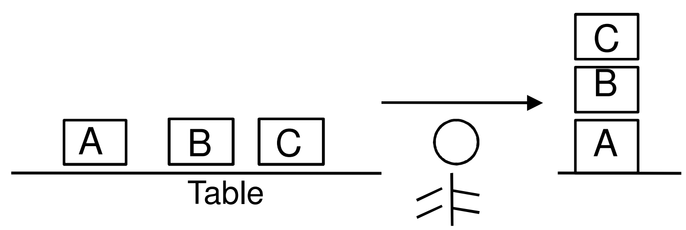
- There is a hypothesis: intelligent action can be reduced to search.
- While intelligence is certainly more than search, the hypothesis is attractive: search is well understood, easily mechanized, manageable, and so on.
- Formally represented knowledge can also be easily used in search.
- Search means systematic traversal of a space of possible solutions of a problem.

The role of search

- A search space is usually a graph (often simple as a tree).
  - A node represents a partial solution.
  - An edge represents a step in the construction of a solution.
- The purpose of search may be:
  - to find a path in the graph from a start node to a goal node (that is, from an initial to a final situation),
  - to find a goal node.

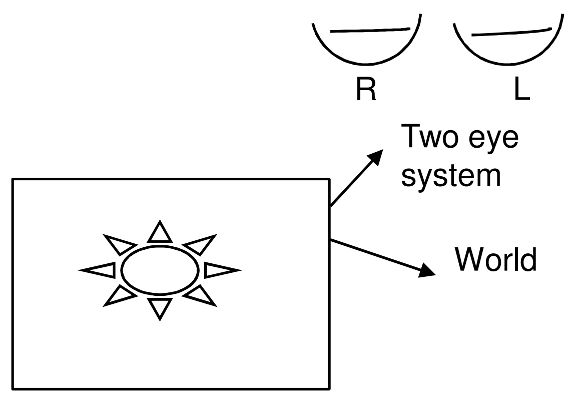
Search Ex1: Planning

- (a) which block to *pick*, (b) which to *stack*, (c) which to *unstack*, (d) whether to *stack* a block or (e) whether to *unstack* an already stacked block. These options have to be searched in order to arrive at the right sequence of actions.



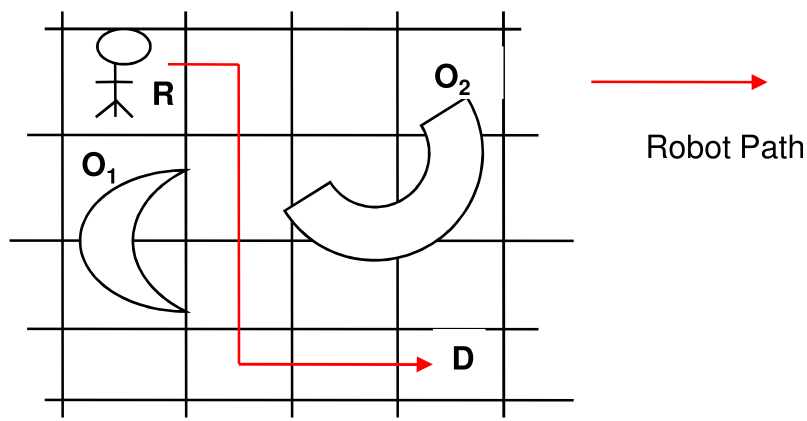
Search Ex2: Vision

- A search needs to be carried out to find which point in the image of *L* corresponds to which point in *R*. Naively carried out, this can become an $O(n^2)$ process where *n* is the number of points in the image.



Search Ex3: Robot Path Planning

- searching amongst the options of moving ***L***eft, ***R***ight, ***U***p or ***D***own. Additionally, each movement has an associated cost representing the relative difficulty of each movement. The search then will have to find the *optimal*, *i.e.*, the *least cost* path.



Search Ex4: Natural Language Processing

- search among many combinations of parts of speech on the way to deciphering the meaning. This applies to every level of processing- *syntax, semantics, pragmatics* and *discourse*.

The man would like to play.

Noun Verb Preposition Verb

Search Ex5: Expert Systems

Search among rules, many of which can apply to a situation:

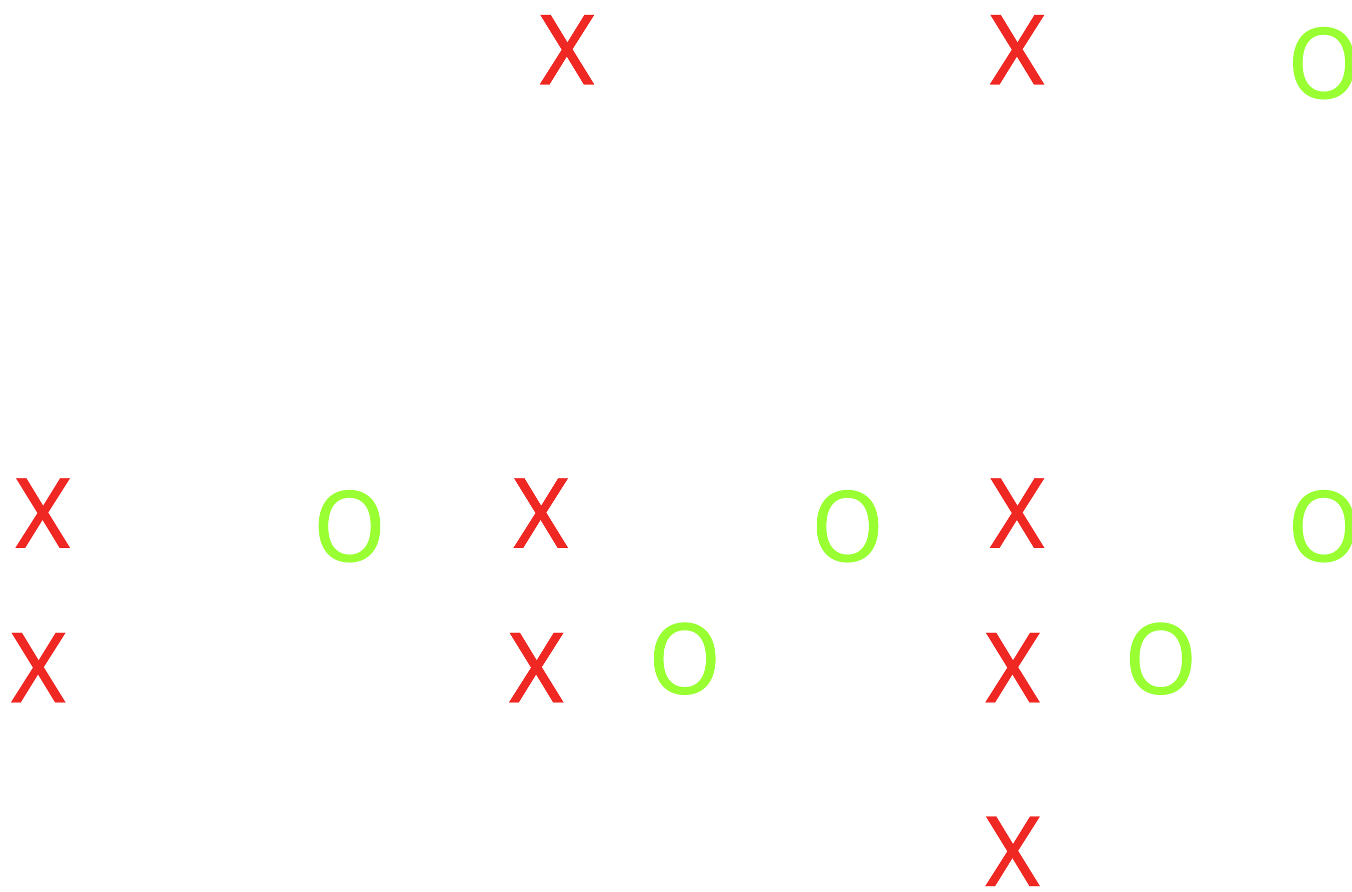
If-conditions

the infection is primary-bacteremia
AND the site of the culture is one of the sterile sites
AND the suspected portal of entry is the gastrointestinal tract
THEN
there is suggestive evidence (0.7) that infection is bacteroid

(from MYCIN)

Search Ex6: Game Playing

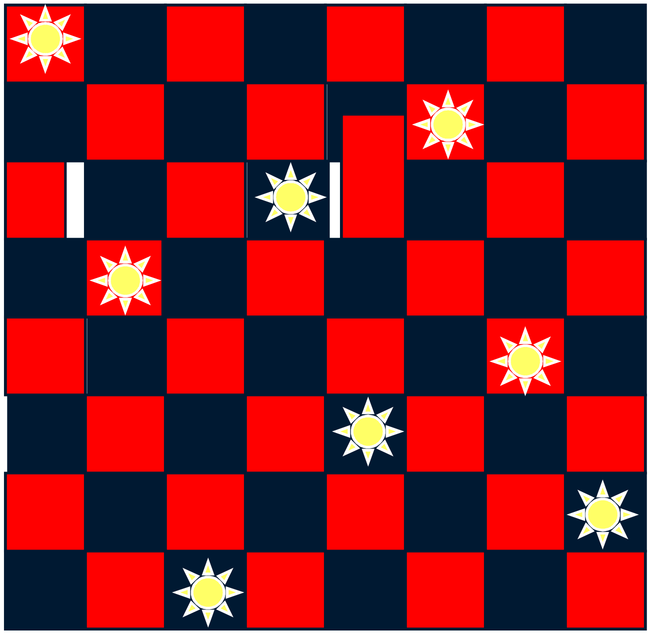
Path to goal isn't quite right.



Search Space Problems

- State Space : Graph of states (Express constraints and parameters of the problem)
- Operators : Transformations applied to the states.
- Start state : S_0 (Search starts from here)
- Goal state(s) : $\{G\}$ - Search terminates here.
- Cost : Effort involved in using an operator.
- Optimal path : Shortest (sometimes least cost) path

8-queen problem



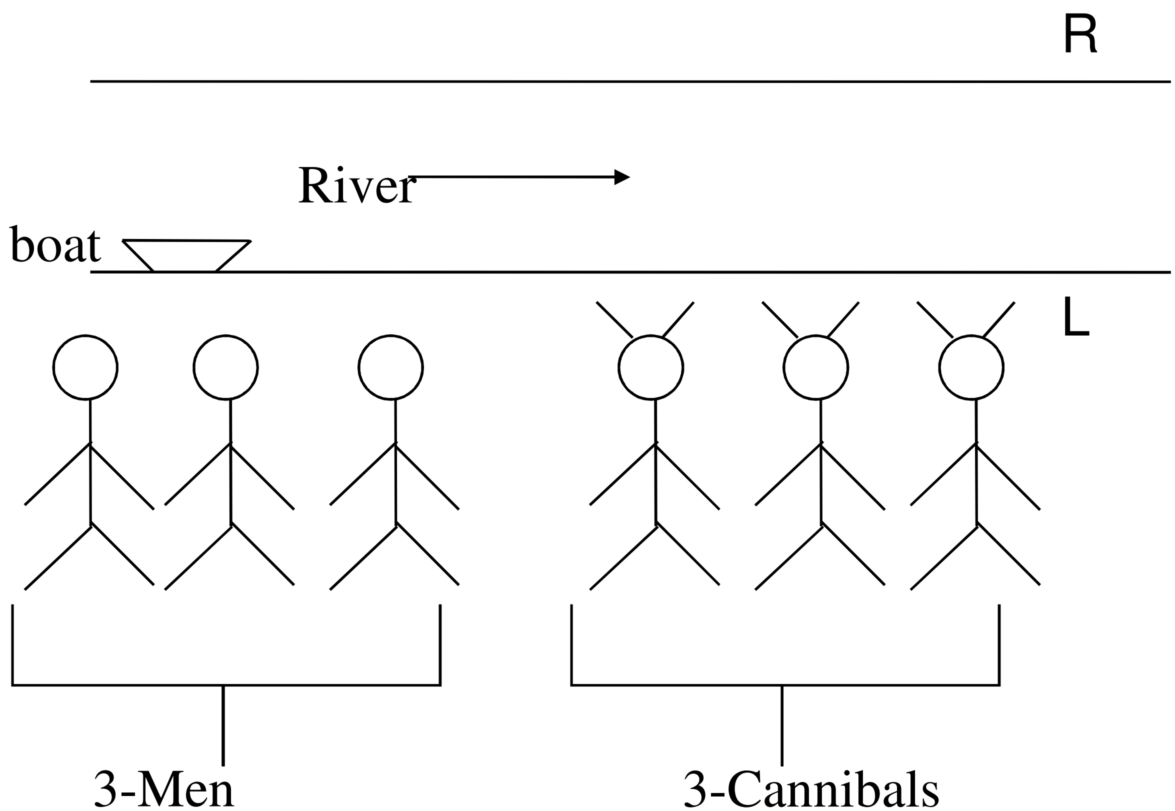
States: Any arrangements of 0 – 8 queens on board

Initial State: No queens on board

Successor function: Add a queen to any empty square

Goal State: 8 queens on the board, unattacked

Men and Cannibals



Constraints

- The boat can carry at most 2 people
- On no bank should the cannibals outnumber the Men
- Move all people to the other side of the river

Men and Cannibals

State : *<#M, #C, P>*

#M = Number of men on bank *L*

#C = Number of cannibals on bank *L*

P = Position of the boat

S0 = *<3, 3, L>*

G = *< 0, 0, R >*

Operations

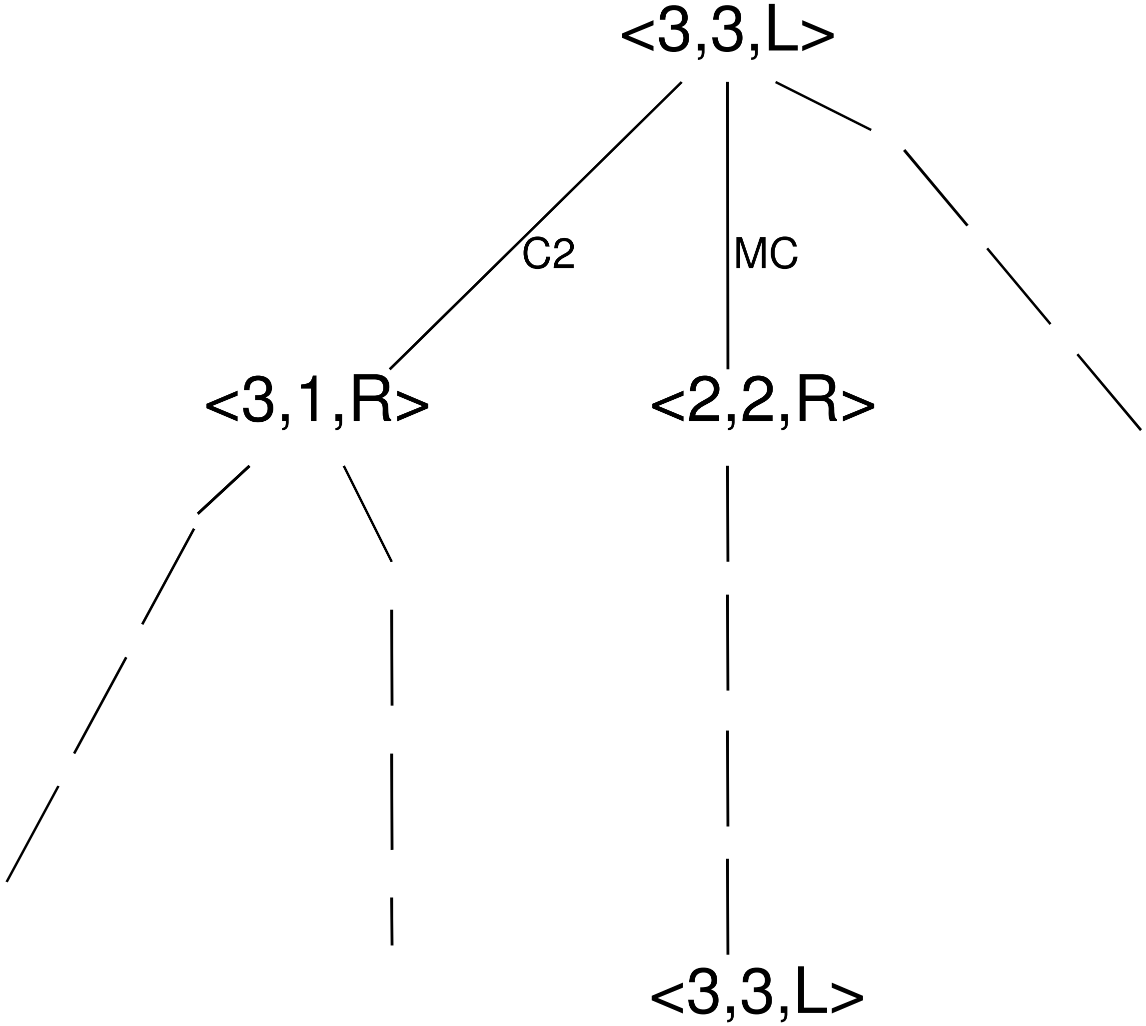
M2 = Two men take boat

M1 = One man takes boat

C2 = Two cannibals take boat

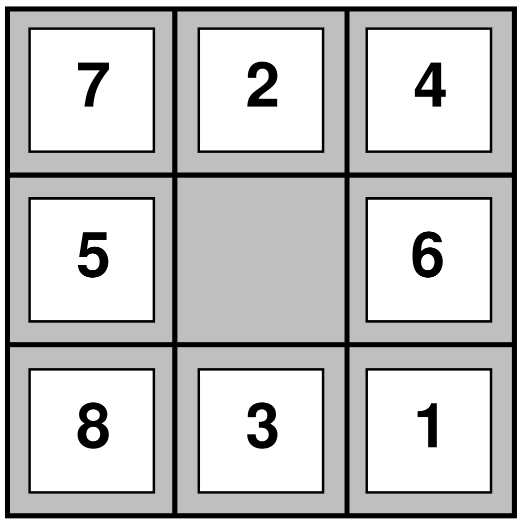
C1 = One cannibal takes boat

MC = One man and one cannibal takes boat

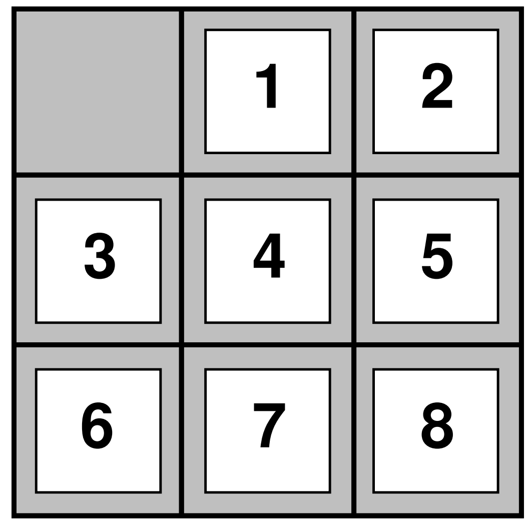


Search tree

8 -puzzle



Start State



Goal State

Standard for the problem... Tile movement represented as the movement of the blank space.

Operators:

L : Blank moves left

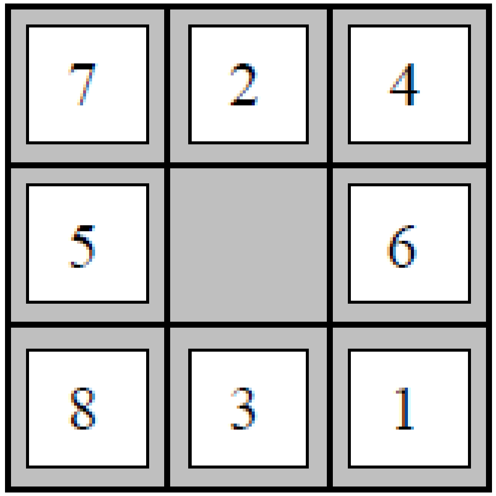
R : Blank moves right

U : Blank moves up

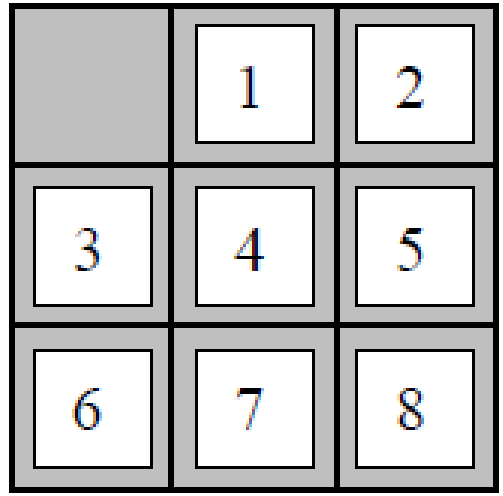
D : Blank moves down

$C(L) = C(R) = C(U) = C(D) = 1$

Example: The 8-puzzle



Start State



Goal State

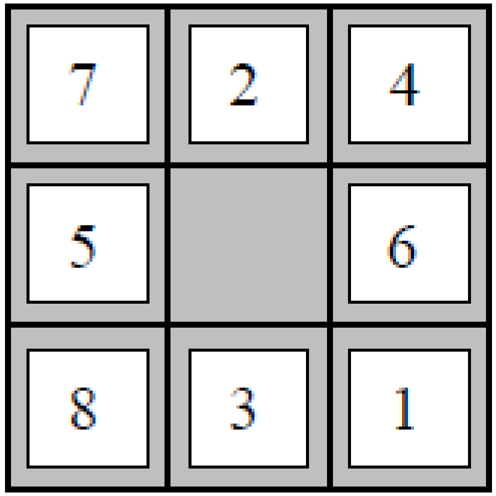
states??

actions??

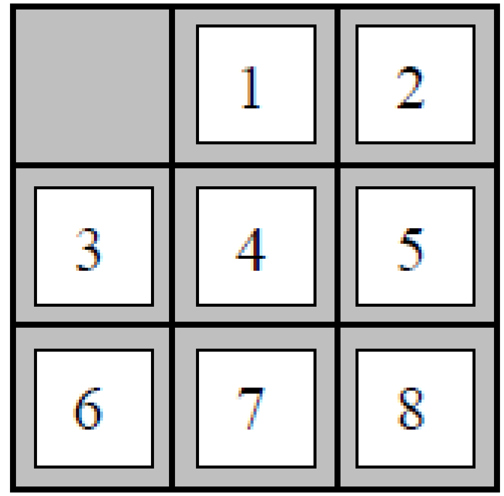
goal test??

path cost??

Example: The 8-puzzle



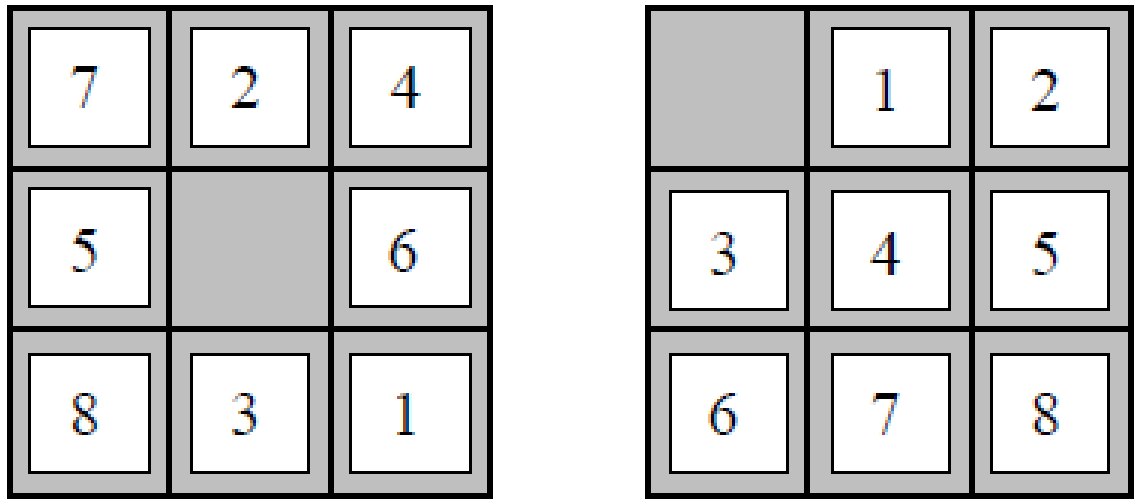
Start State



Goal State

states??: integer locations of tiles (ignore intermediate positions)
actions??
goal test??
path cost??

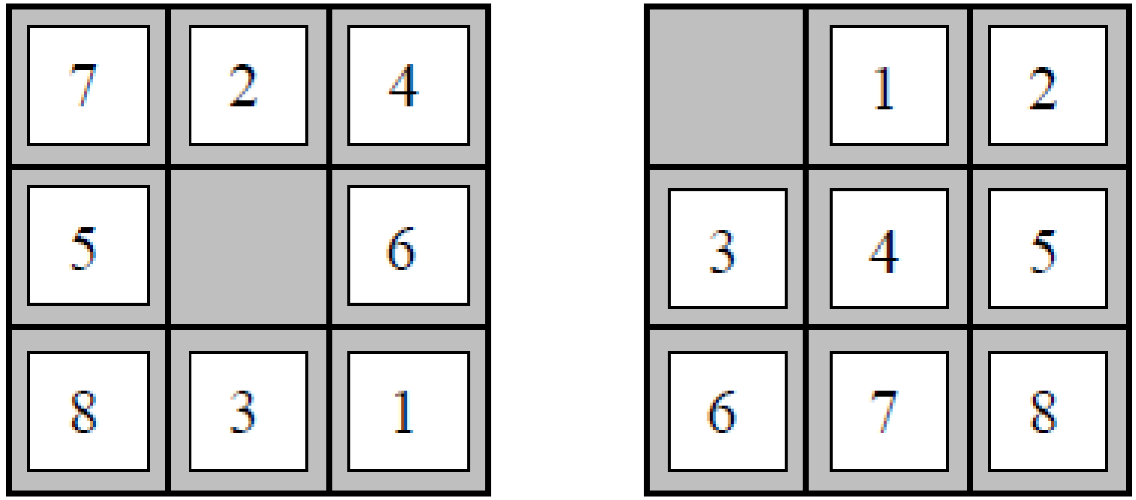
Example: The 8-puzzle



Start State   Goal State

states??: integer locations of tiles (ignore intermediate positions)
actions??: move blank left, right, up, down (ignore unjamming etc.)
goal test??
path cost??

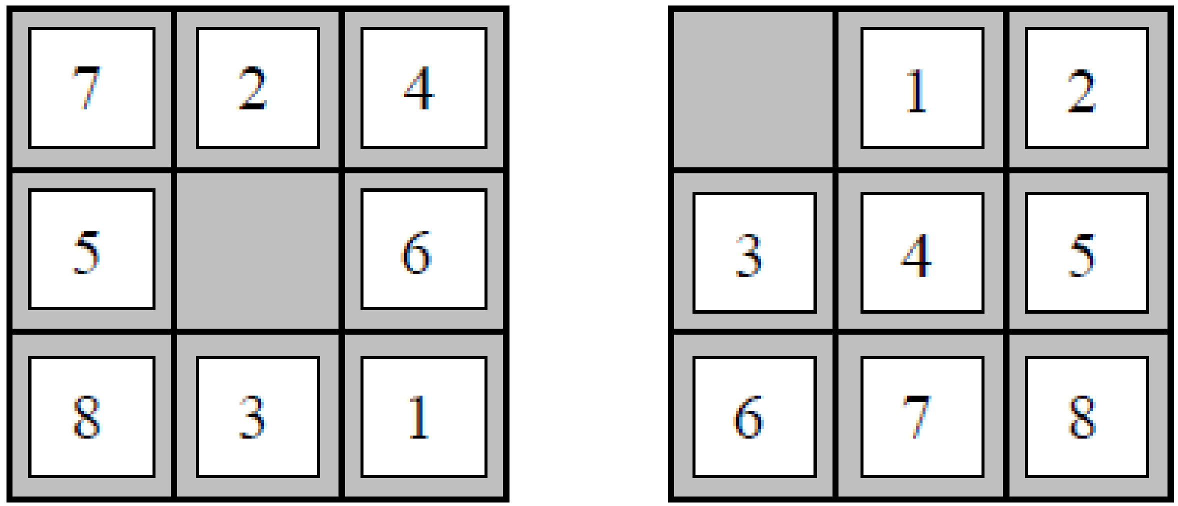
Example: The 8-puzzle



Start State Goal State

states??: integer locations of tiles (ignore intermediate positions)
actions??: move blank left, right, up, down (ignore unjamming etc.)
goal test??: = goal state (given)
path cost??

Example: The 8-puzzle

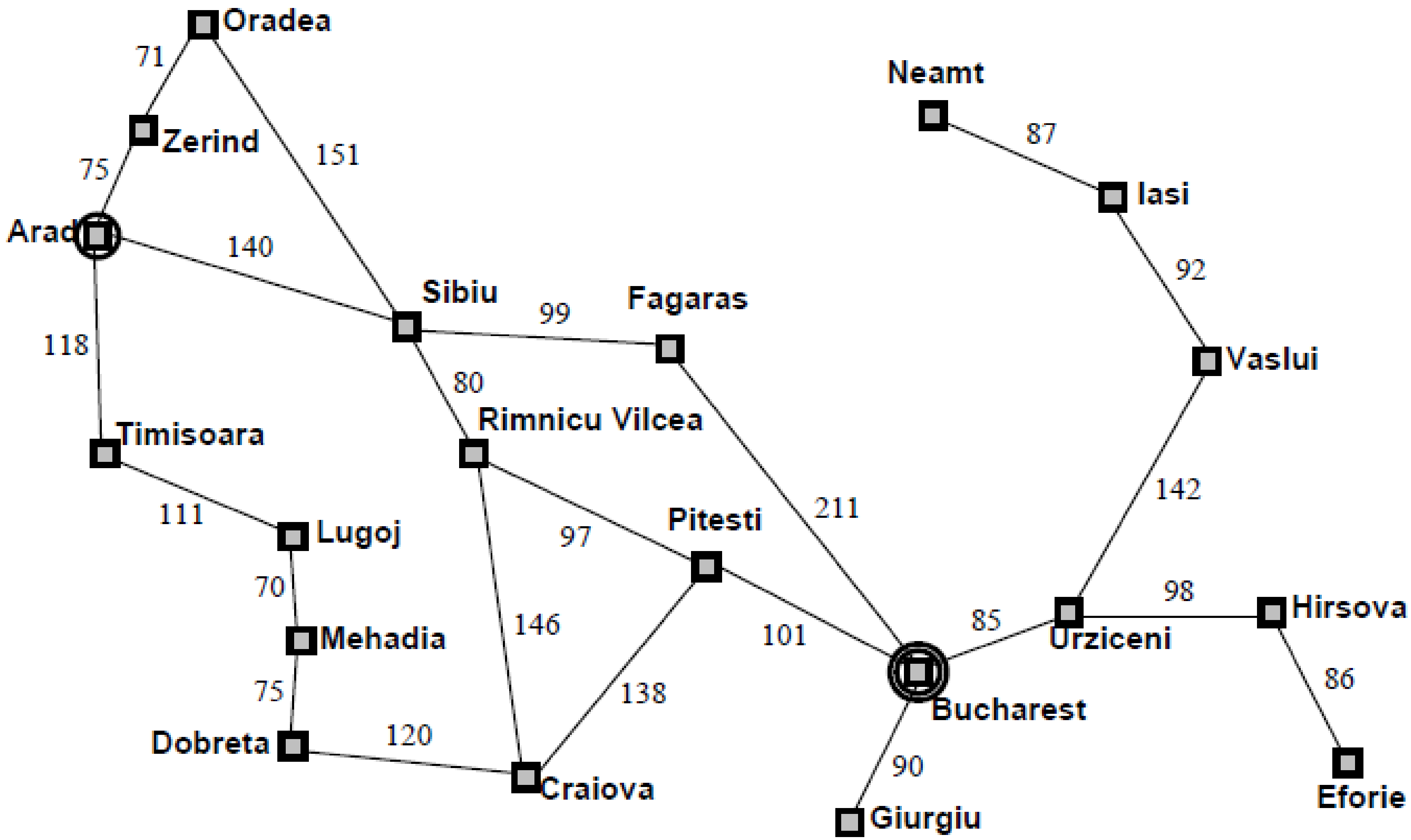


Start State Goal State

states??: integer locations of tiles (ignore intermediate positions)
actions??: move blank left, right, up, down (ignore unjamming etc.)
goal test??: = goal state (given)
path cost??: 1 per move

[Note: optimal solution of n-Puzzle family is NP-hard]

Example: Romania



Example: Romania

On holiday in Romania; currently in Arad.
Flight leaves tomorrow from Bucharest

Formulate goal:
be in Bucharest

Formulate problem:
states: various cities
actions: drive between cities

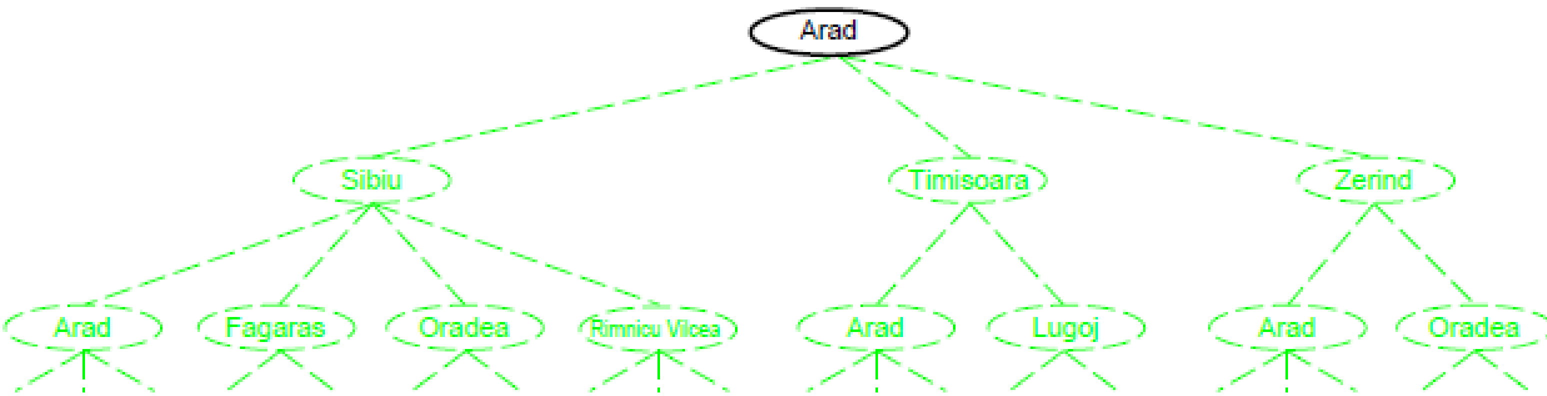
Find solution:
sequence of cities, e.g., Arad, Sibiu, Fagaras, Bucharest

Example: Romania

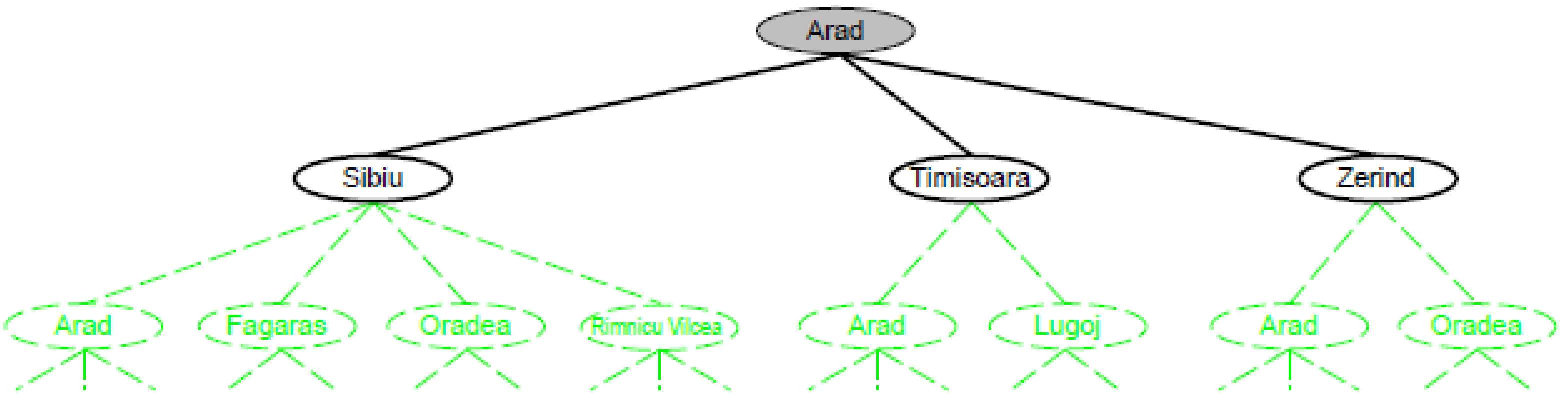
- Initial State: at Arad
- States Space: being at any city
- Successor function: set of state-pairs

  S(Arad) = Zerind
- Goal State: Bucharest
- Path cost : sum of distances
- Solution : sequence of states

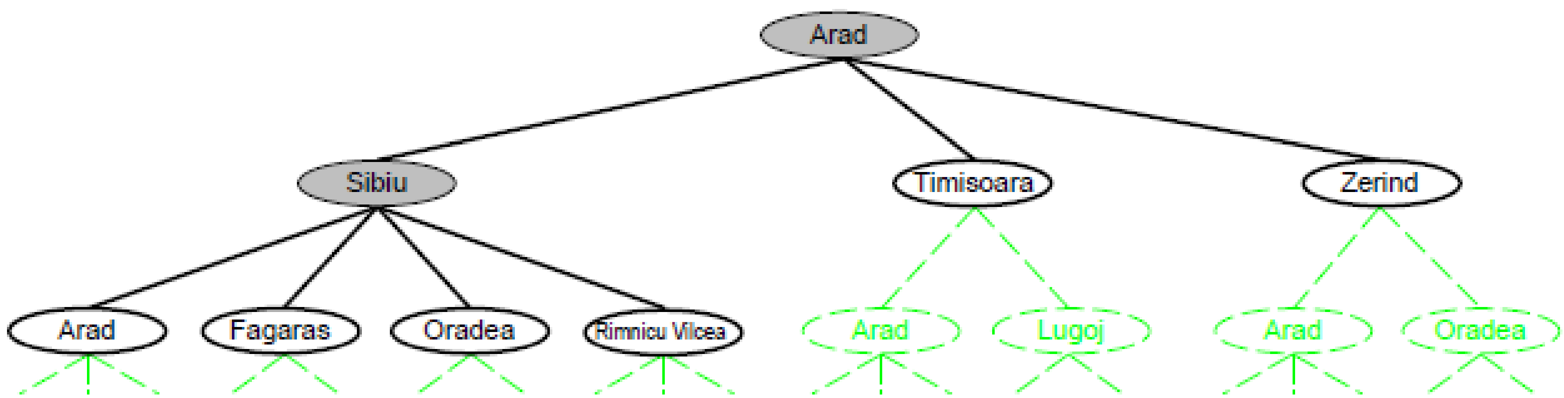
Example: Romania



Example: Romania



Example: Romania

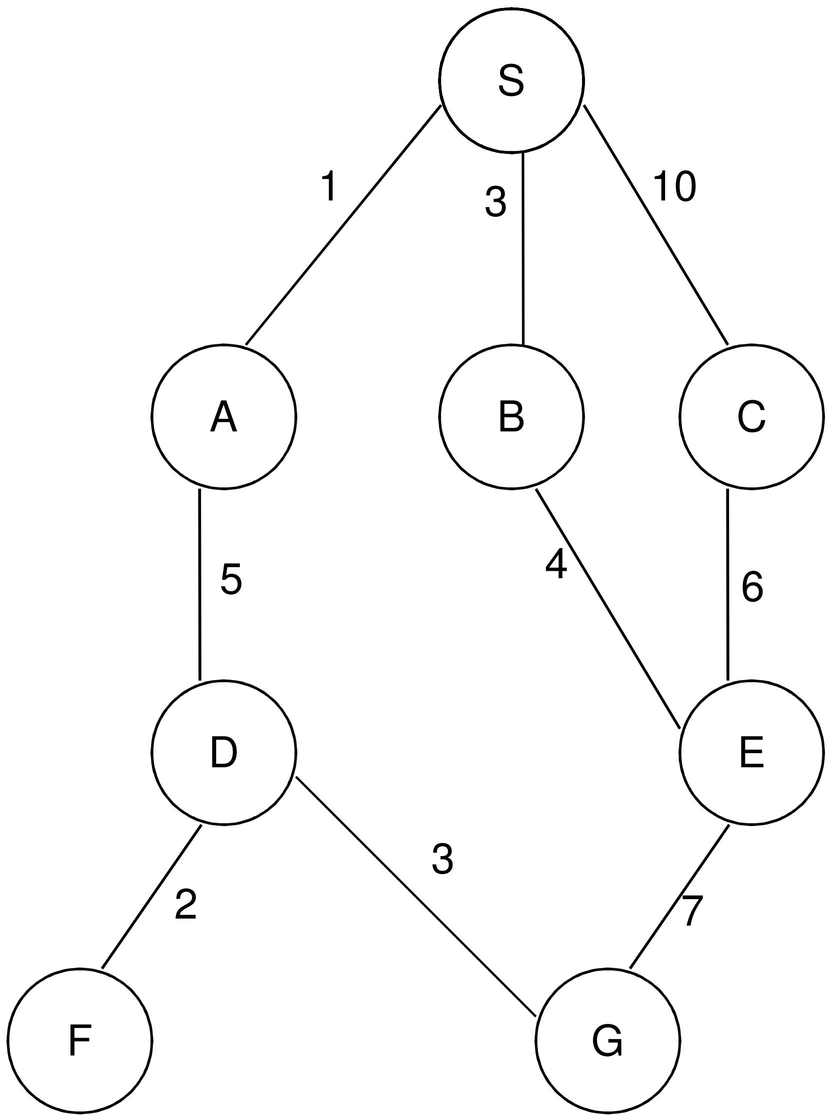


Tree Search Algorithm

- Basic Idea: Simulated exploration of state space by generating successors of already explored

```
function TREE-SEARCH(problem, strategy) returns a solution, or failure
    initialize the search tree using the initial state of problem
    loop do
        if there are no candidates for expansion then return failure
        choose a leaf node for expansion according to strategy
        if the node contains a goal state then return the corresponding solution
        else expand the node and add the resulting nodes to the search tree
    end
```

General Graph search Algorithm



Graph G = (V,E)

1) Open List : S $^{(\emptyset, 0)}$

Closed list : Ø

2) OL : $A^{(S,1)}$, $B^{(S,3)}$, $C^{(S,10)}$

CL : S

3) OL : $B^{(S,3)}$, $C^{(S,10)}$, $D^{(A,6)}$

CL : S, A

4) OL : $C^{(S,10)}$, $D^{(A,6)}$, $E^{(B,7)}$

CL: S, A, B

5) OL : $D^{(A,6)}$, $E^{(B,7)}$

CL : S, A, B , C

6) OL : $E^{(B,7)}$, $F^{(D,8)}$, $G^{(D, 9)}$

CL : S, A, B, C, D

7) OL : $F^{(D,8)}$, $G^{(D,9)}$

CL : S, A, B, C, D, E

8) OL : $G^{(D,9)}$

CL : S, A, B, C, D, E, F

9) OL : Ø

CL : S, A, B, C, D, E,
F, G

Steps of GGS

1. Create a search graph *G*, consisting only of the start node *S*; put *S* on a list called *OPEN.*

2. Create a list called *CLOSED* that is initially empty.

3. Loop: if *OPEN* is empty, exit with failure.

4. Select the first node on *OPEN*, remove from *OPEN* and put on *CLOSED*, call this node *n*.

5. if *n* is the goal node, exit with the solution obtained by tracing a path along the pointers from *n* to *s* in *G*.

6. Expand node *n*, generating the set *M* of its successors that are not ancestors of *n*.

GGS steps (contd.)

7. Establish a pointer to *n* from those members of *M* that were not already in *G* (*i.e.*, not already on either *OPEN* or *CLOSED*). Add these members of *M* to *OPEN*. For each member of *M* that was already on *OPEN* or *CLOSED*, decide whether or not to redirect its pointer to *n*. For each member of M already on *CLOSED*, decide for each of its descendents in *G* whether or not to redirect its pointer.

8. Reorder the list *OPEN* using some strategy.

9. Go *LOOP*.

Search Strategies

Uninformed/Blind Search

- Breadth First Search
- Depth First Search
- Depth Limited Search
- Bidirectional Search

Informed/Heuristic Search

- Hill Climbing Search
- A* Algorithm