Mansoura University
Faculty of Computers and Information
Department of Computer Science
First Semester: 2020-2021

[CS324P] Artificial Intelligence - 1 : Solving Problems By Searching

Grade: Third Year (Computer Science)

Ass. Prof. Taher Hamza

Dr. Sara El-Metwally

Faculty of Computers and Information,

Mansoura University,

Egypt.

Contents

1 Problem formulation

2 Search strategies

3 Un-informed search

4 Applications

Problem Formulation

- A problem is defined by five items:

1. Initial state
2. Actions
3. Transition model (Successor)
4. Goal test
5. Path cost

Problem Formulation

- State space is a the set of all states reachable from the initial state by any sequence of actions.

- The state space forms a directed network or graph in which the nodes are states and the links between nodes are actions.

- Path is a sequence of states connected by a sequence of actions.

- Search is a process of looking for a sequence of actions that reach the goal.

- Solution is a sequence of actions leading from the initial state to a goal state.

Problem Formulation

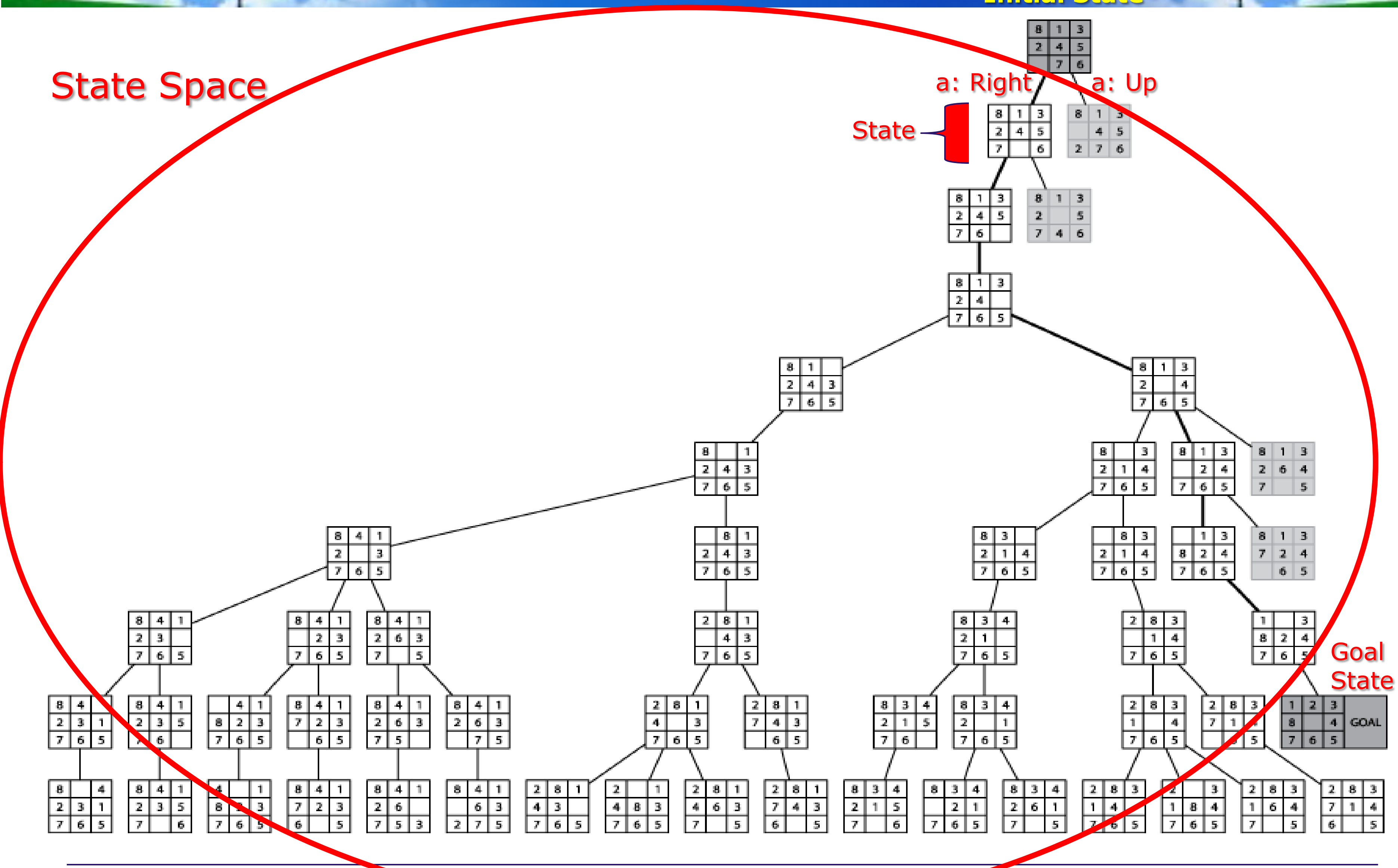
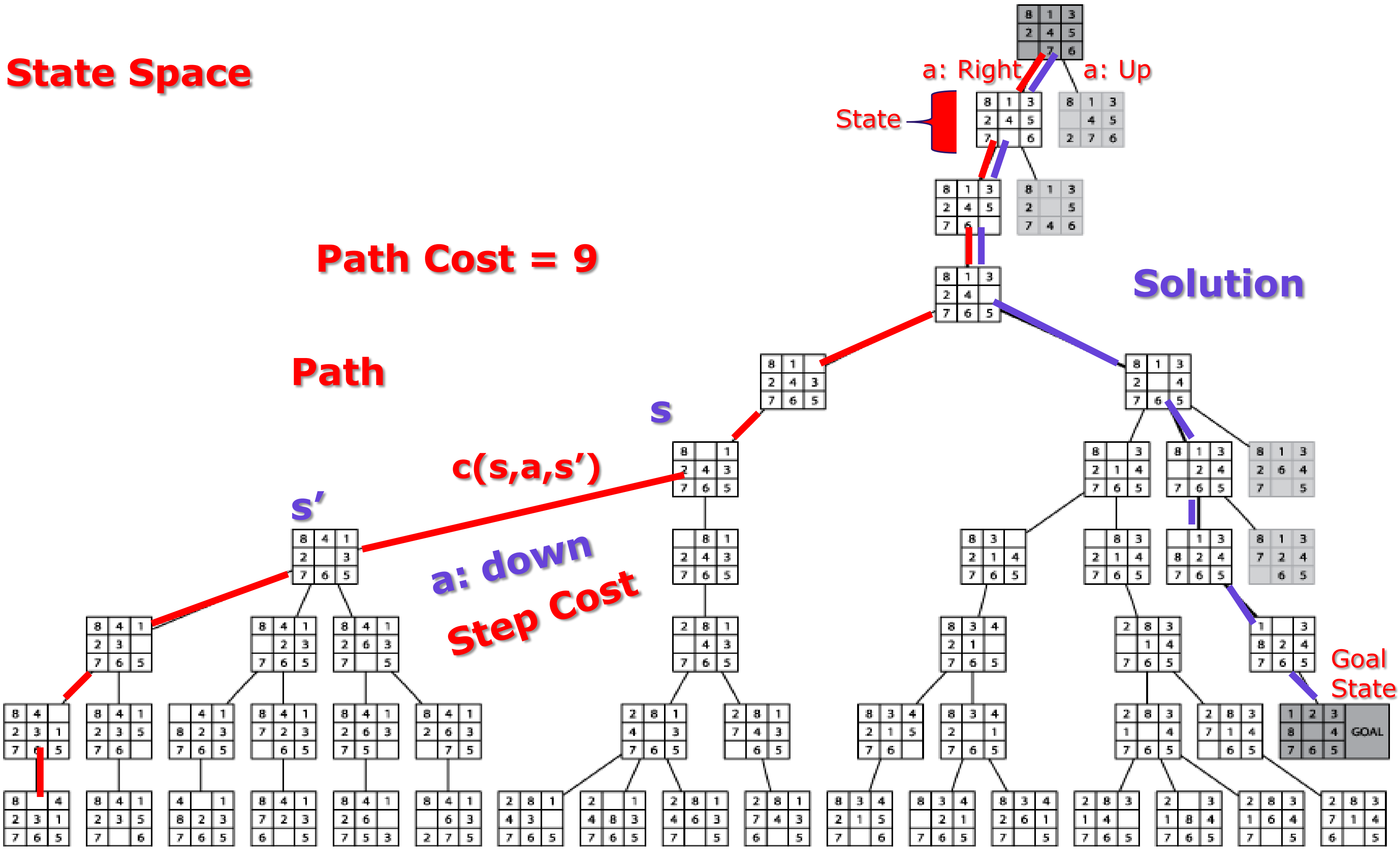


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Problem Formulation



Initial State

State Space

a: Right

a: Up

State

Path Cost = 9

Solution

Path

s

c(s,a,s')

s'

a: down

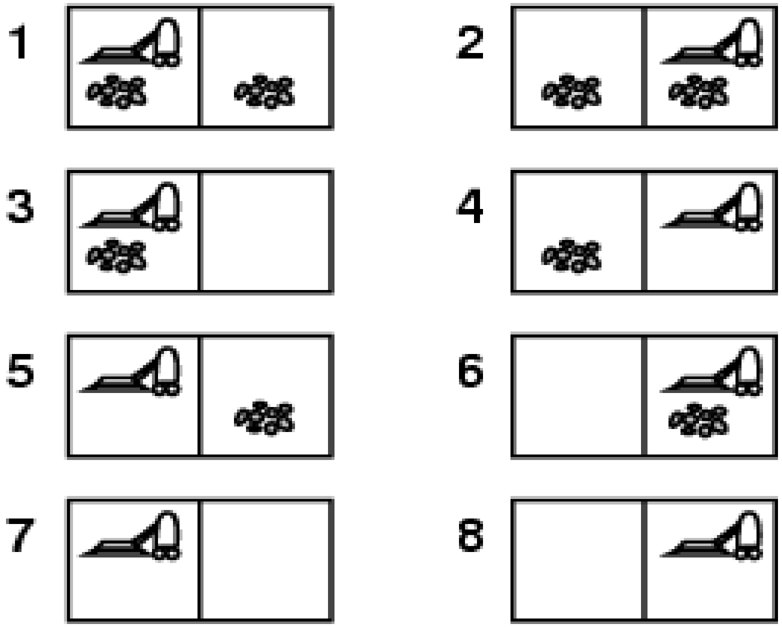
Step Cost

Goal State

Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Problem Formulation, example

Vacuum- cleaner

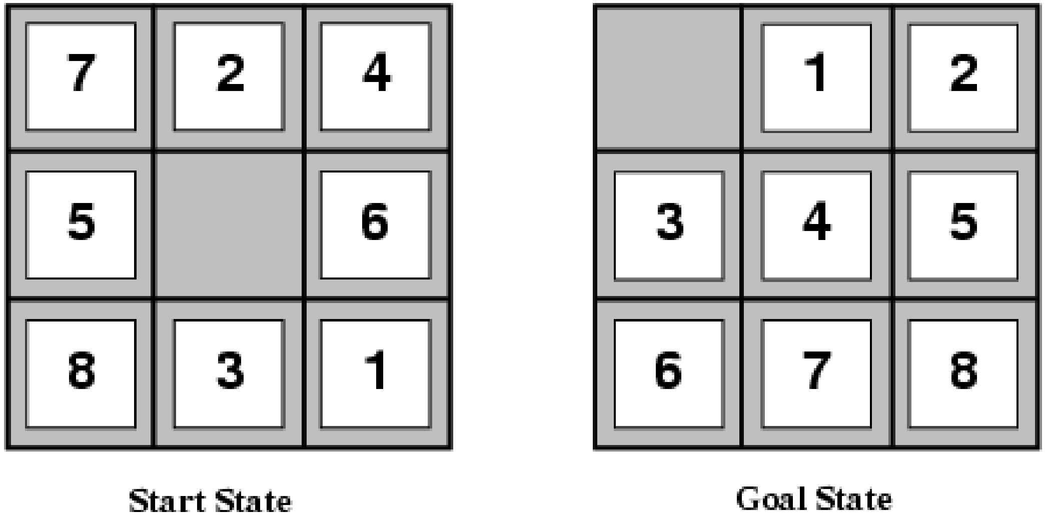


- **States:** agent locations and square status
- **Initial State:** any random state
- **Successor function:** **left**, **right**, **Suck**
- **Goal test:** All squares are clean
- **Path cost:** 1 per step

Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Problem Formulation, example

N-puzzle



Start State

7	2	4
5		6
8	3	1

Goal State

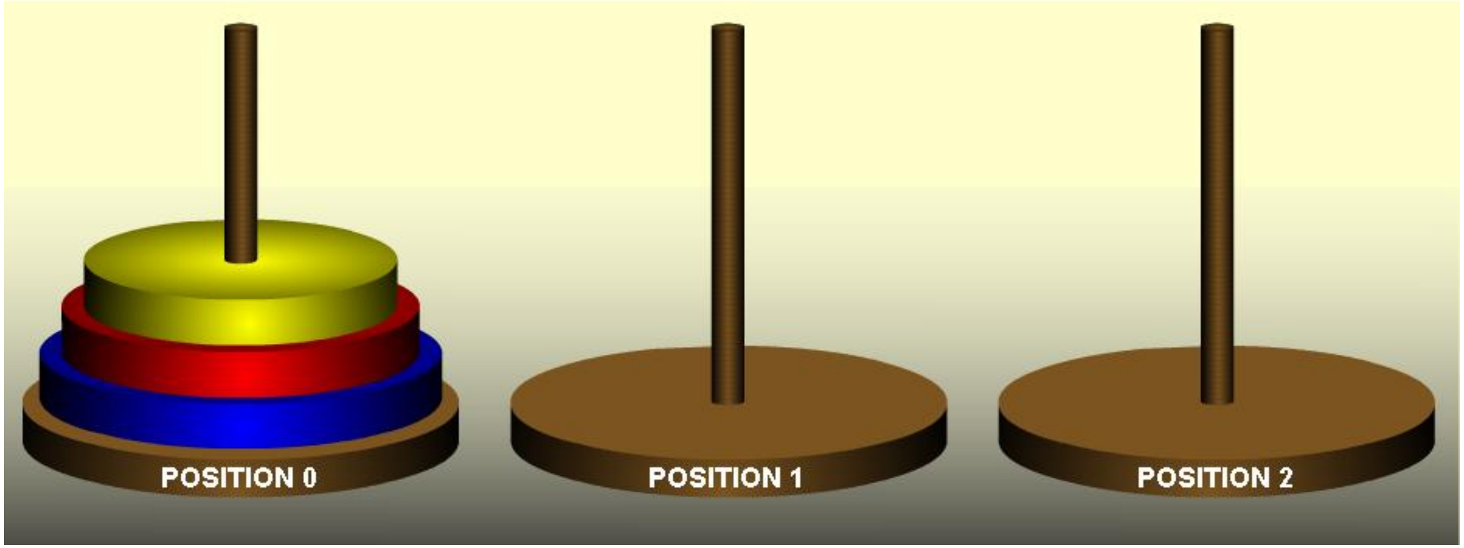
	1	2
3	4	5
6	7	8

- **States:** locations of tiles
- **Initial State:** any random arrangement
- **Successor function:** move blank **left**, **right**, **up**, **down**
- **Goal test:** Ordered arrangement
- **Path cost:** 1 per move

Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Problem Formulation, example

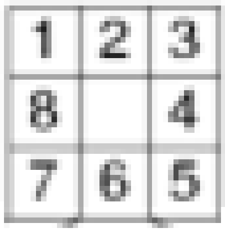
Tower of Hanoi



- **States:** disks location in the three possible positions
- **Initial State:** All disks in position 0
- **Successor function:** move disk between positions (with constraints)
- **Goal test:** All disks in position 2
- **Path cost:** 1 per move

Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

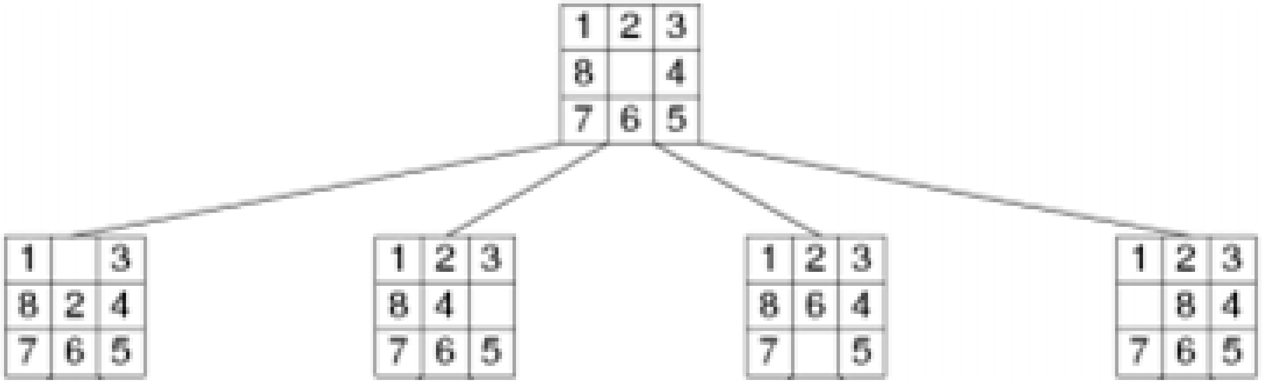
Search Space



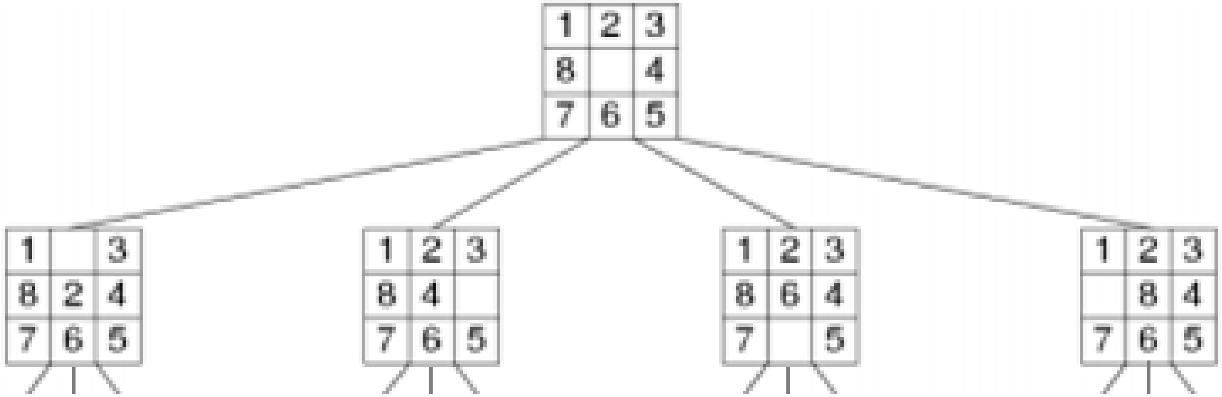
1	2	3
8		4
7	6	5

Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Search Space



Search Space



A search strategy is defined by picking the order of node expansion

Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Search Space

Vacuum- cleaner

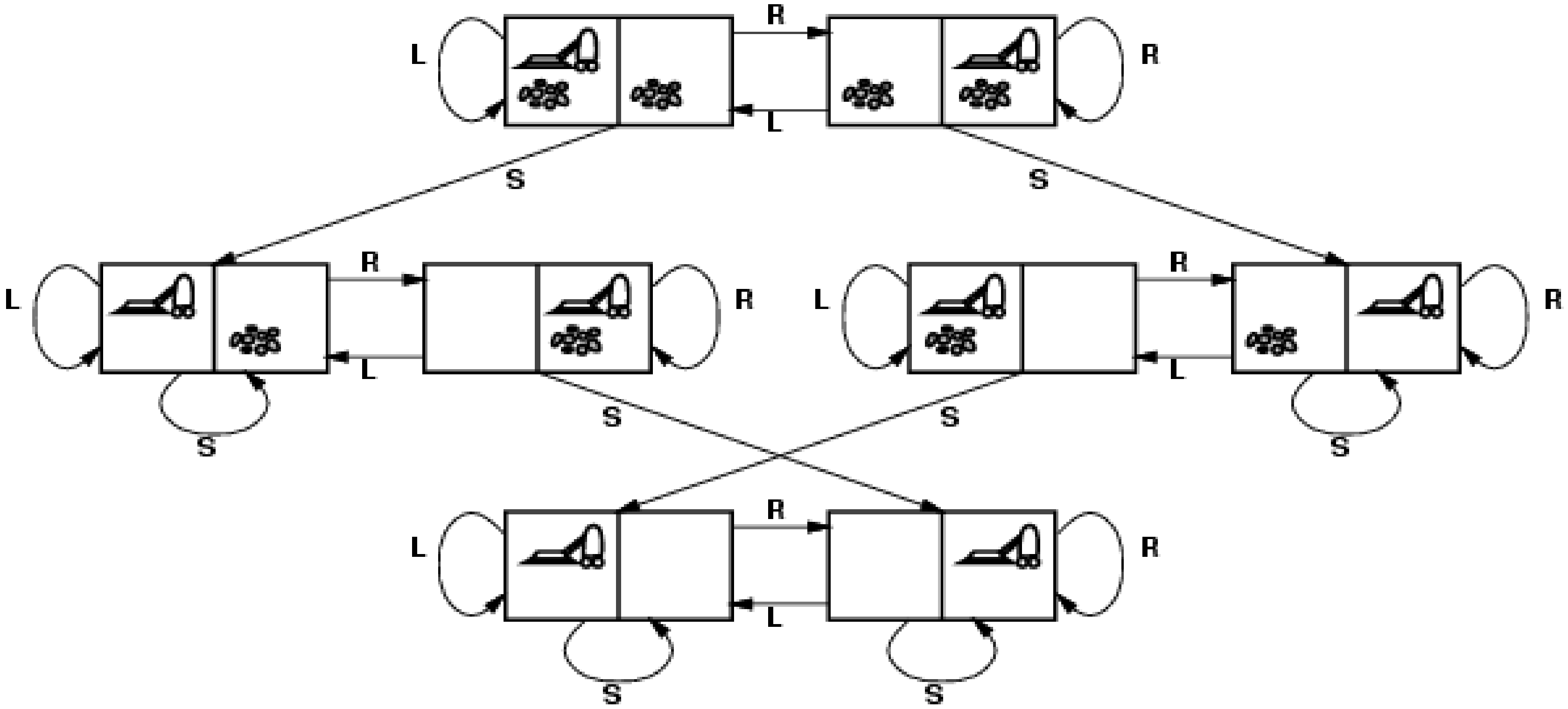
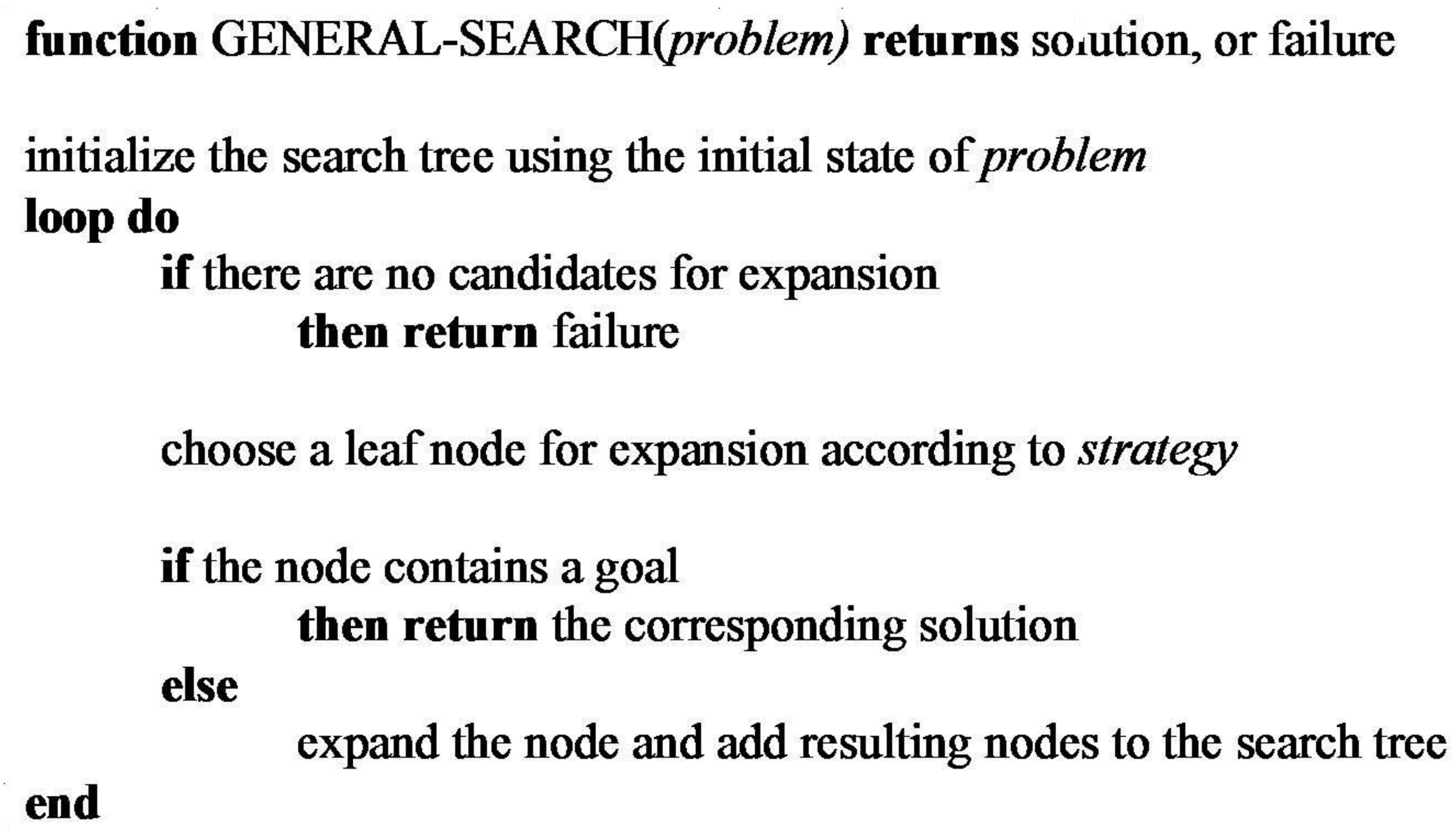


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Search strategy



```
function GENERAL-SEARCH(problem) returns soıution, 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
                then return the corresponding solution
        else
                expand the node and add resulting nodes to the search tree
end
```

Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Search strategy

- States vs. Nodes

  - State is a (representation of) a physical configuration
  - Node is a data structure constituting part of a search tree includes state, parent node, action, path cost g(n), depth

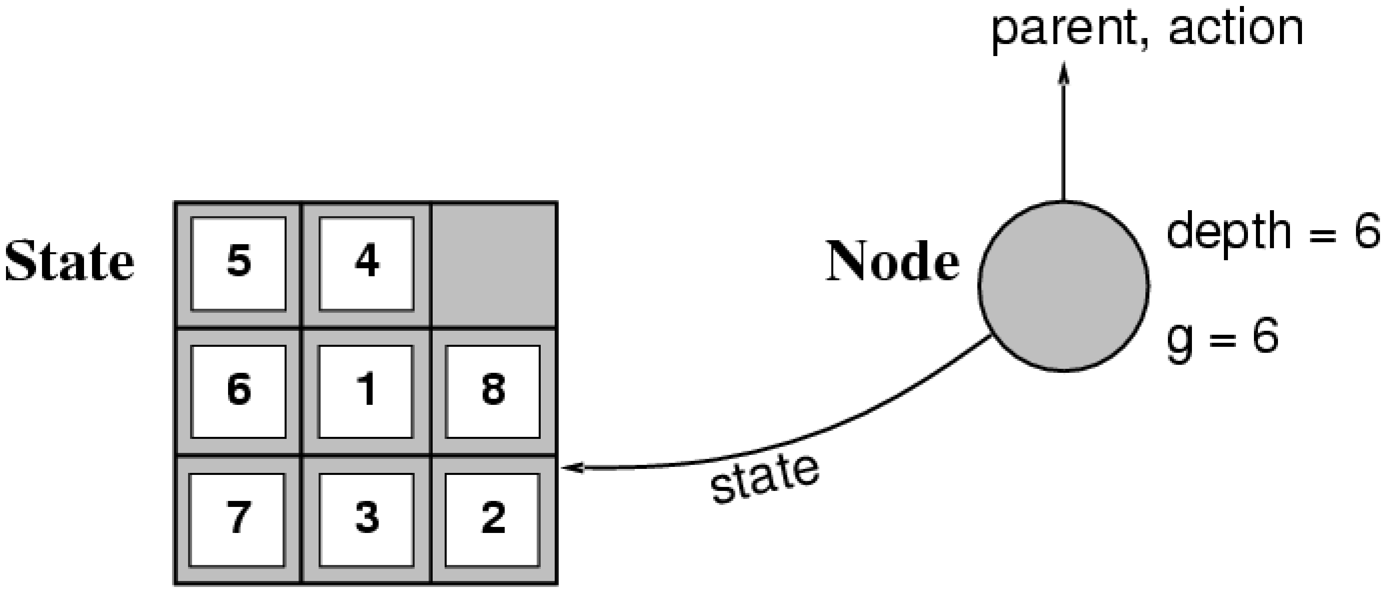
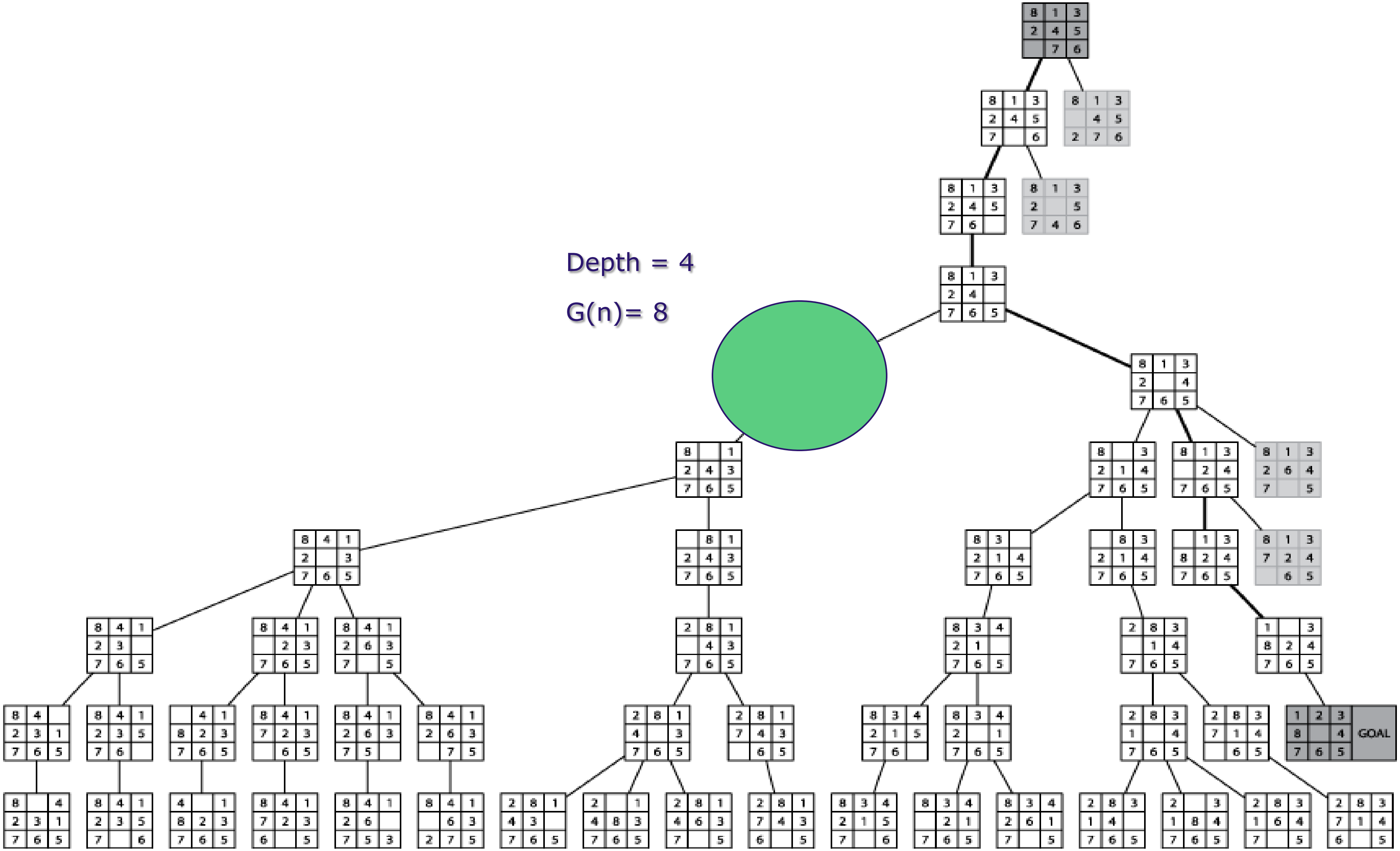


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Search strategy



Depth = 4

G(n)= 8

Search strategy

- Strategies are evaluated along the following properties:

Completeness

Guaranteed to find a solution when there is one?

Optimality

Finds the optimal solution?

Time

How long does it take to find a solution?

Space

How much memory is needed to perform the search?

- **b:** maximum branching factor of the search tree
- **d:** depth of the least-cost (optimum) solution
- **m:** maximum depth of the state space

Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Uninformed (blind) Search

- Uninformed search strategies use only the information available in the problem definition

- Uninformed search algorithms:

  1. Breadth-first search (BFS)
  2. Uniform-cost search (UCS)
  3. Depth-first search (DFS)
  4. Depth-limited search (DLS)
  5. Iterative deepening search (IDS)
  6. Bidirectional search(BS)

Blind Search

1. Breadth-first search (BFS)

- Expand shallowest unexpanded node
- Nodes are stored in FIFO queue (new successors go at end)

Queue: [A]

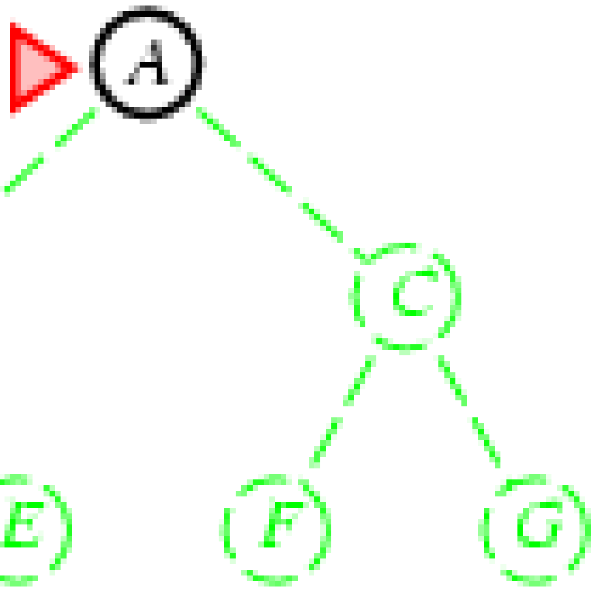


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

1. Breadth-first search (BFS)

- Expand shallowest unexpanded node
- Nodes are stored in FIFO queue (new successors go at end)

Queue: [B, C]

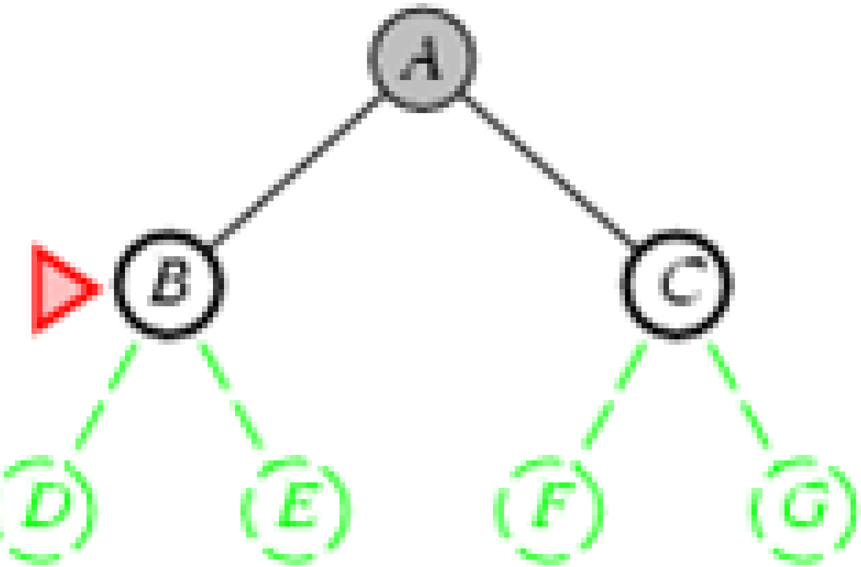


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

1. Breadth-first search (BFS)

- Expand shallowest unexpanded node
- Nodes are stored in FIFO queue (new successors go at end)

Queue: [C,D, E]

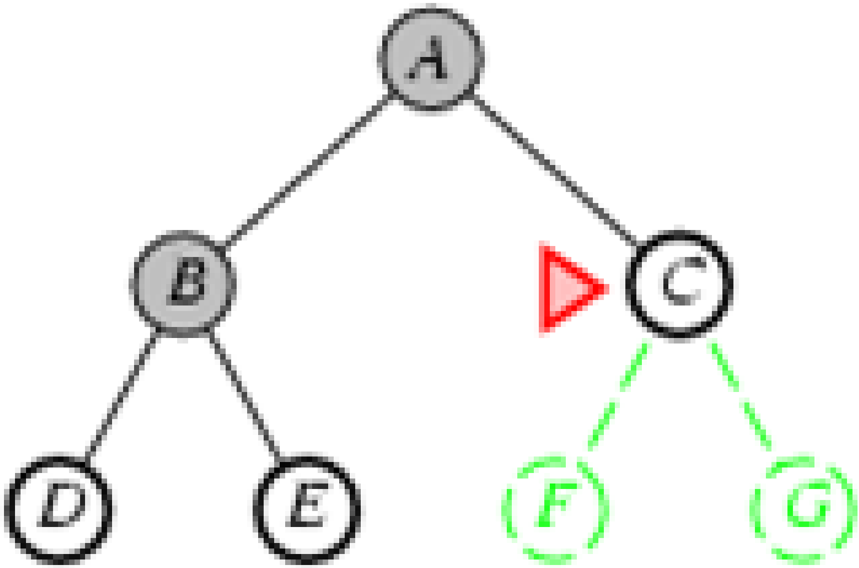


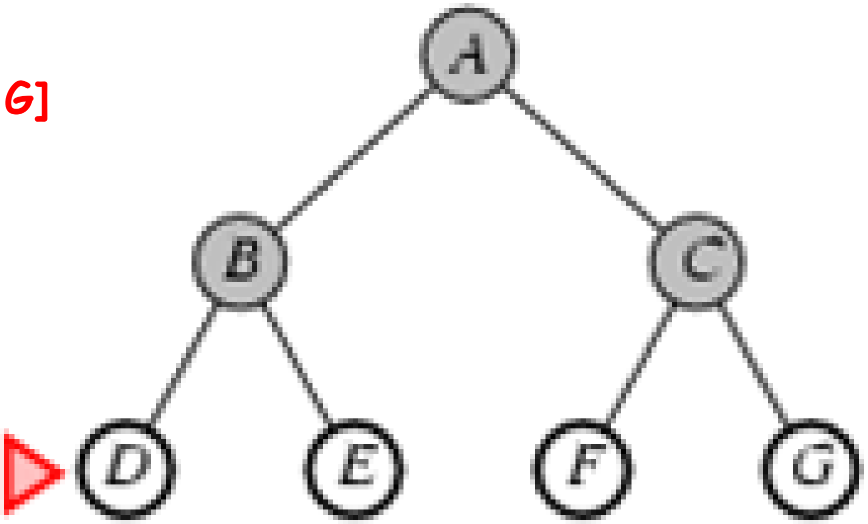
Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

1. Breadth-first search (BFS)

- Expand shallowest unexpanded node
- Nodes are stored in FIFO queue (new successors go at end)

Queue: [D, E,F,G]

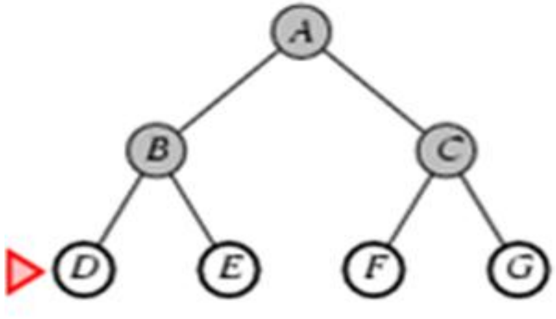


Goal is found!

Image Credit: Artificial Intelligence A Modern Approach Second Edition by Stuart J. Russell and Peter Norvig by 2010.

Blind Search, evaluation

1. Breadth-first search (BFS)



- Complete? Yes (if b is finite)
- Time? $O(b^d)$
- Space? $O(b^d)$
- Optimal? Yes (if all trans. have same cost)

Space is the main problem

Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

2. Uniform-cost search (UCS)

- Expand least-cost unexpanded node
- Nodes are stored in Ordered queue (order by cost)

Consider this state space for a given problem :

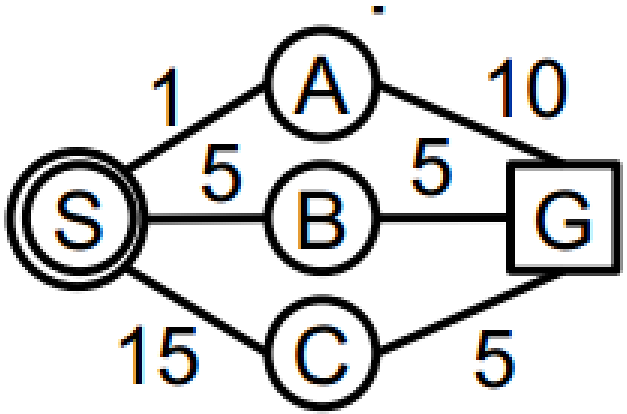
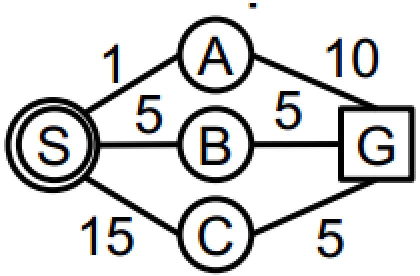


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

2. Uniform-cost search (UCS), example

- Expand least-cost unexpanded node
- Nodes are stored in Ordered queue (order by cost)



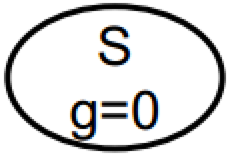
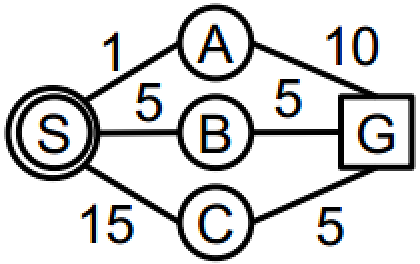


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

2. Uniform-cost search (UCS), example

- Expand least-cost unexpanded node
- Nodes are stored in Ordered queue (order by cost)



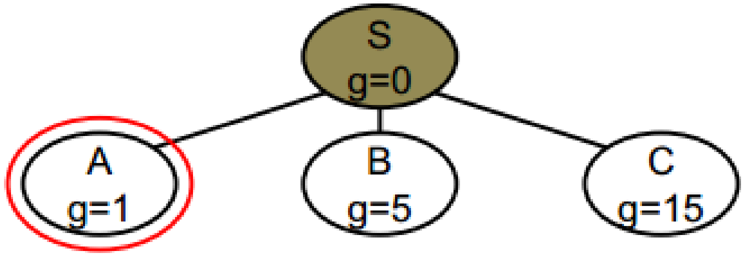
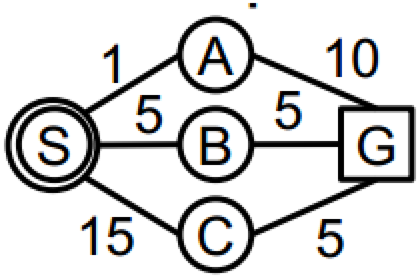


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

2. Uniform-cost search (UCS), example

- Expand least-cost unexpanded node
- Nodes are stored in Ordered queue (order by cost)



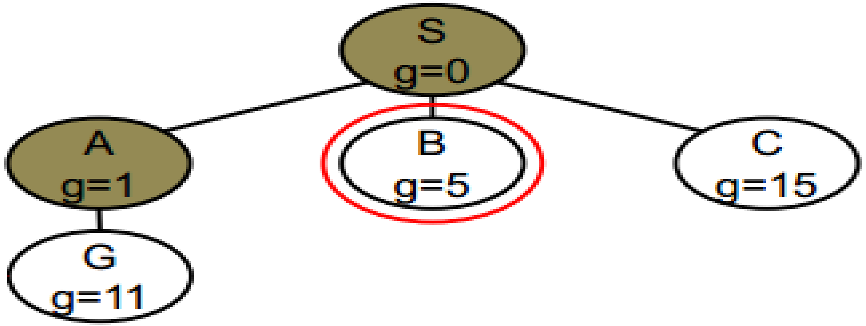
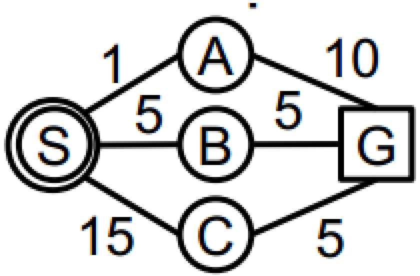


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

2. Uniform-cost search (UCS), example

- Expand least-cost unexpanded node
- Nodes are stored in Ordered queue (order by cost)



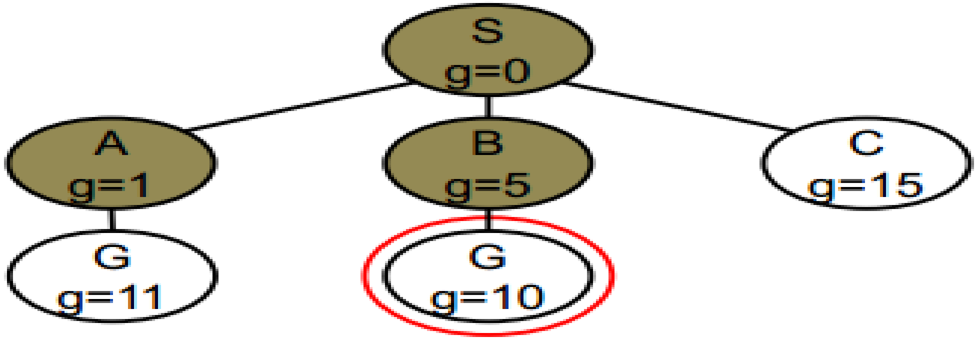
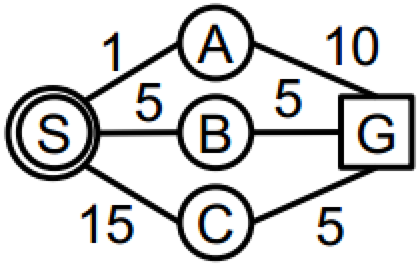


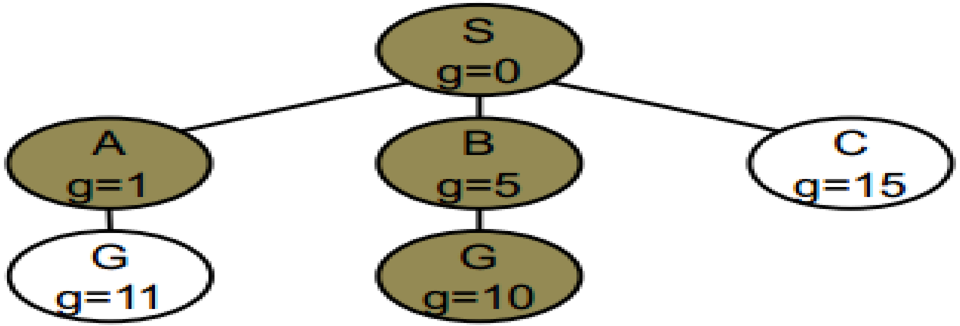
Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

2. Uniform-cost search (UCS), example

- Expand least-cost unexpanded node
- Nodes are stored in Ordered queue (order by cost)





Optimum goal is found!

Blind Search, evaluation

2. Uniform-cost search (UCS)

- Complete? Yes (if *g*> €)
- Time? # of nodes with *g* ≤ cost of optimal solution
- Space? # of nodes with *g* ≤ cost of optimal solution
- Optimal? Yes

Equivalent to BFS if step costs all equal

Blind Search

3. Depth-first search (DFS)

- Expand deepest unexpanded node
- Nodes are stored in LIFO stack(put successors at front)

Stack: [A]

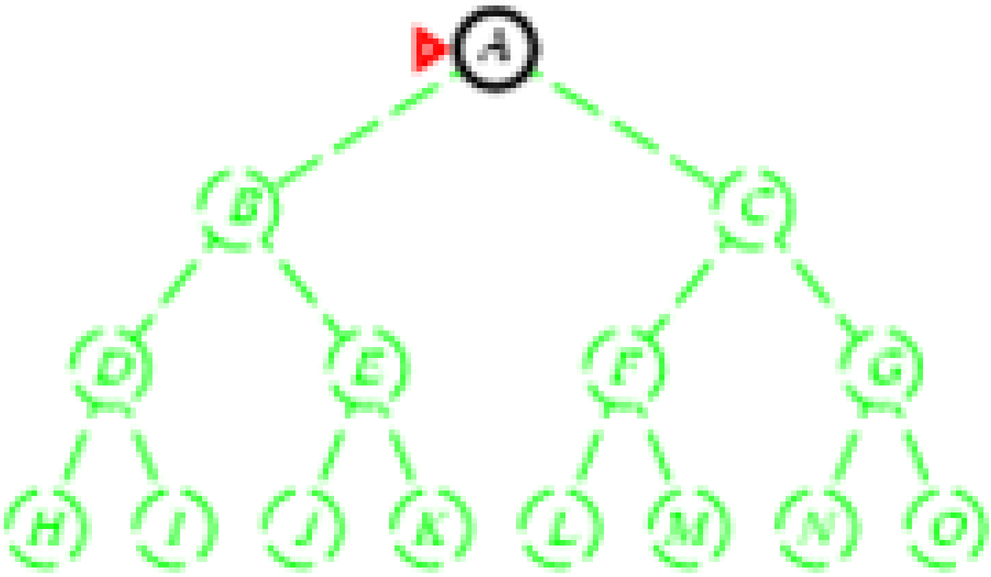


Image Credit: Artificial Intelligence A Modern Approach Second Edition by Stuart J. Russell and Peter Norvig by 2010.

Blind Search

3. Depth-first search (DFS)

- Expand deepest unexpanded node
- Nodes are stored in LIFO stack (put successors at front)

Stack: [B,C]

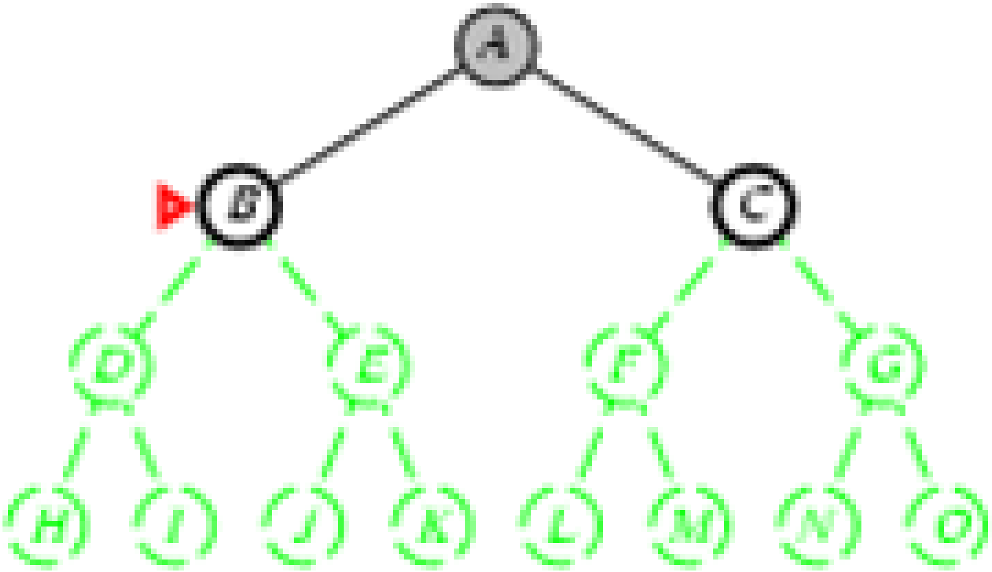


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

3. Depth-first search (DFS)

- Expand deepest unexpanded node
- Nodes are stored in LIFO stack (put successors at front)

Stack: [D,E,C]

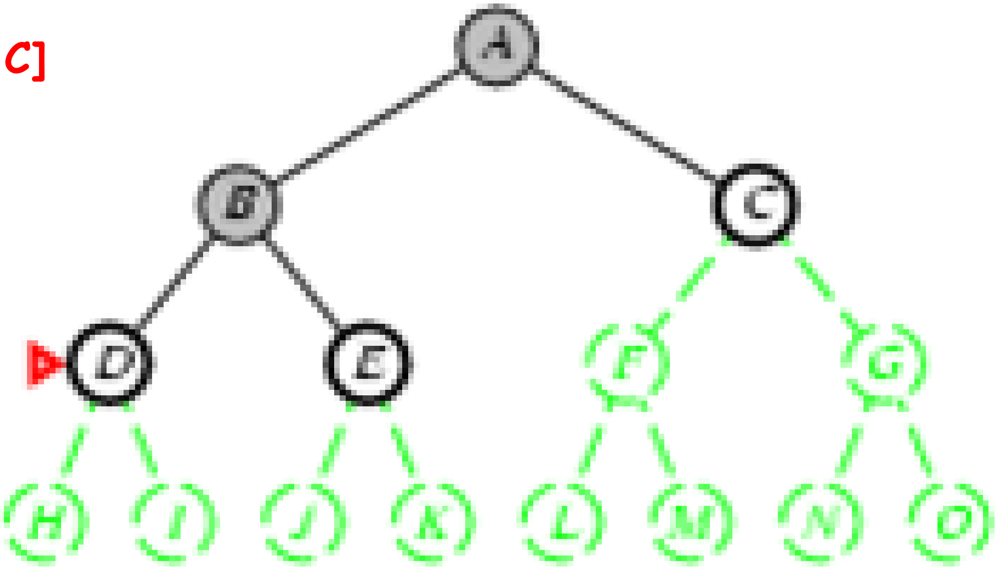


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

3. Depth-first search (DFS)

- Expand deepest unexpanded node
- Nodes are stored in LIFO stack (put successors at front)

Stack: [H,I,E,C]

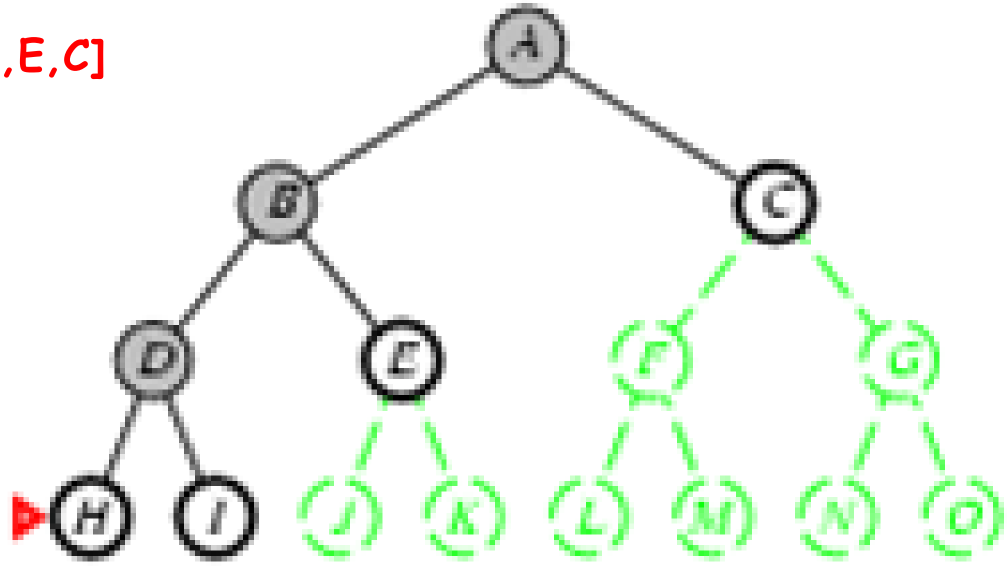


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

3. Depth-first search (DFS)

- Expand deepest unexpanded node
- Nodes are stored in LIFO stack (put successors at front)

Stack: [I,E,C]

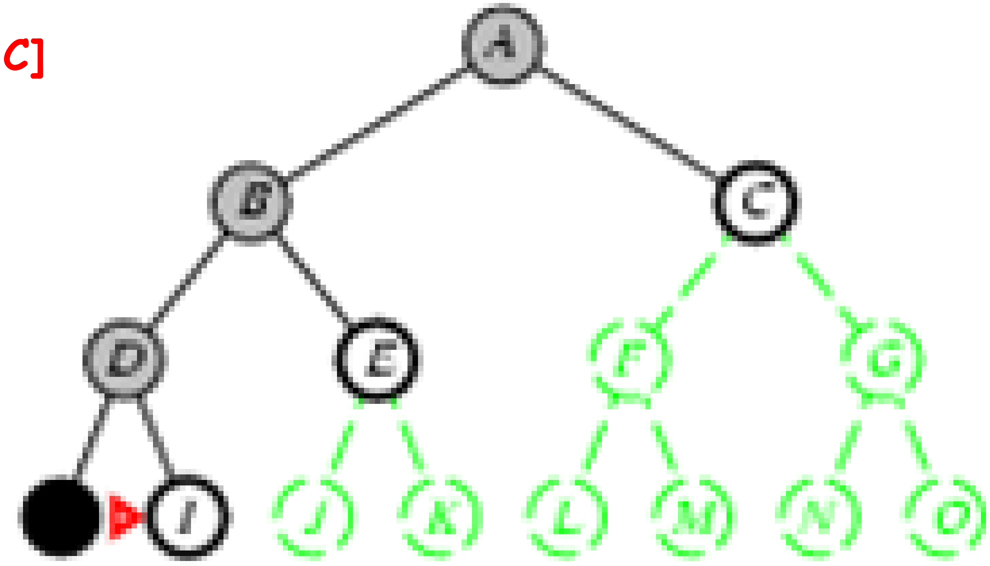


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

3. Depth-first search (DFS)

- Expand deepest unexpanded node
- Nodes are stored in LIFO stack (put successors at front)

Stack: [E,C]

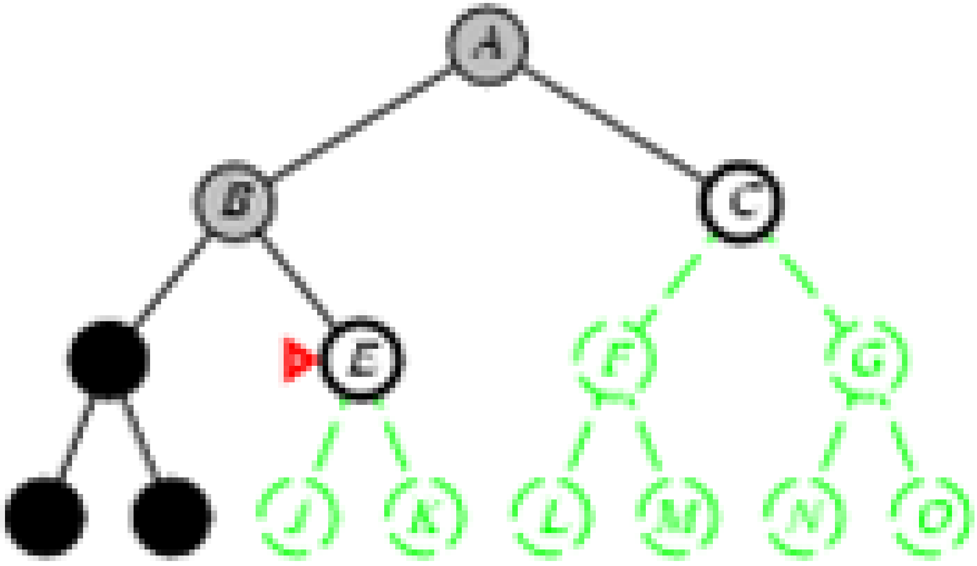


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

3. Depth-first search (DFS)

- Expand deepest unexpanded node
- Nodes are stored in LIFO stack (put successors at front)

Stack: [J,K,C]

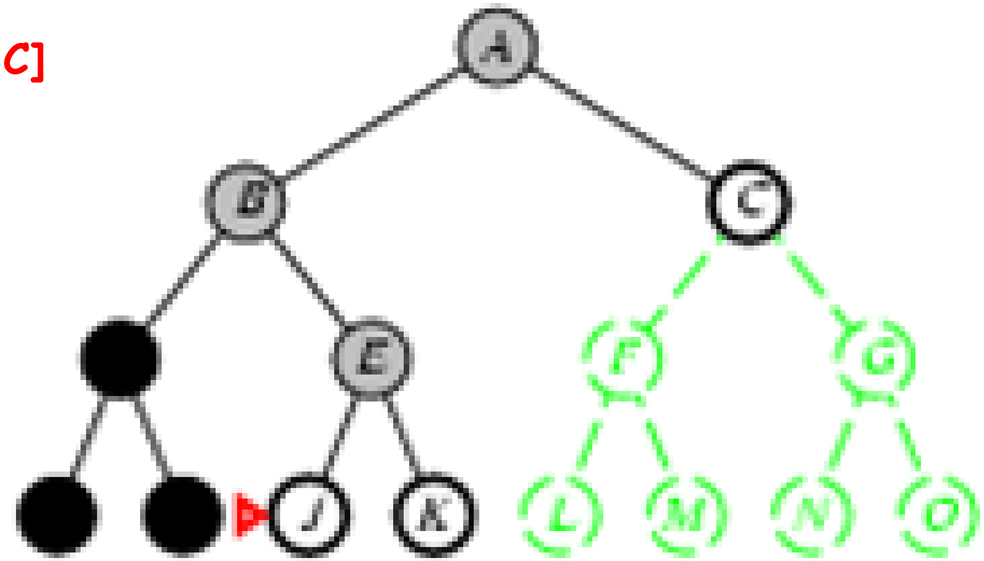


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

3. Depth-first search (DFS)

- Expand deepest unexpanded node
- Nodes are stored in LIFO stack (put successors at front)

Stack: [K,C]

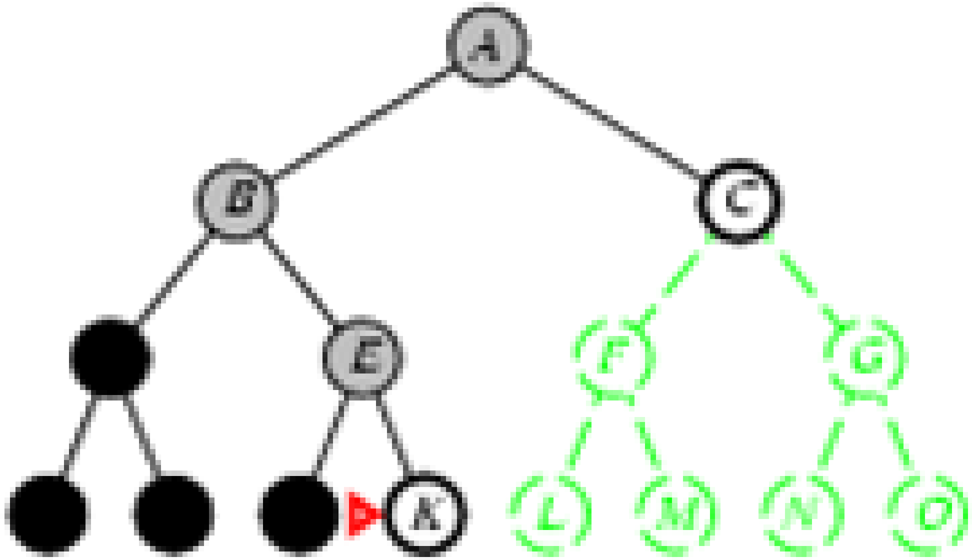


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

3. Depth-first search (DFS)

- Expand deepest unexpanded node
- Nodes are stored in LIFO stack (put successors at front)

Stack: [C]

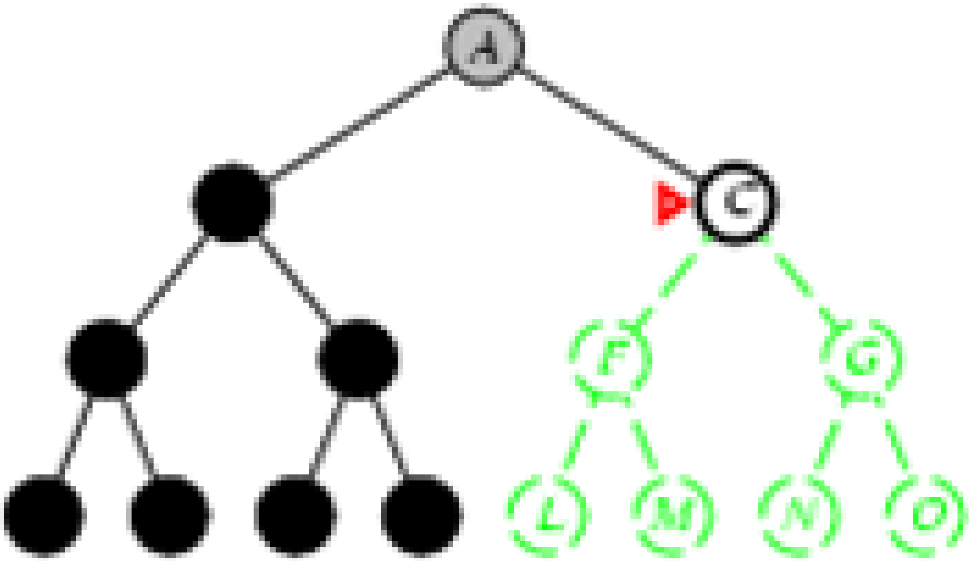


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

3. Depth-first search (DFS)

- Expand deepest unexpanded node
- Nodes are stored in LIFO stack (put successors at front)

Stack: [F,G]

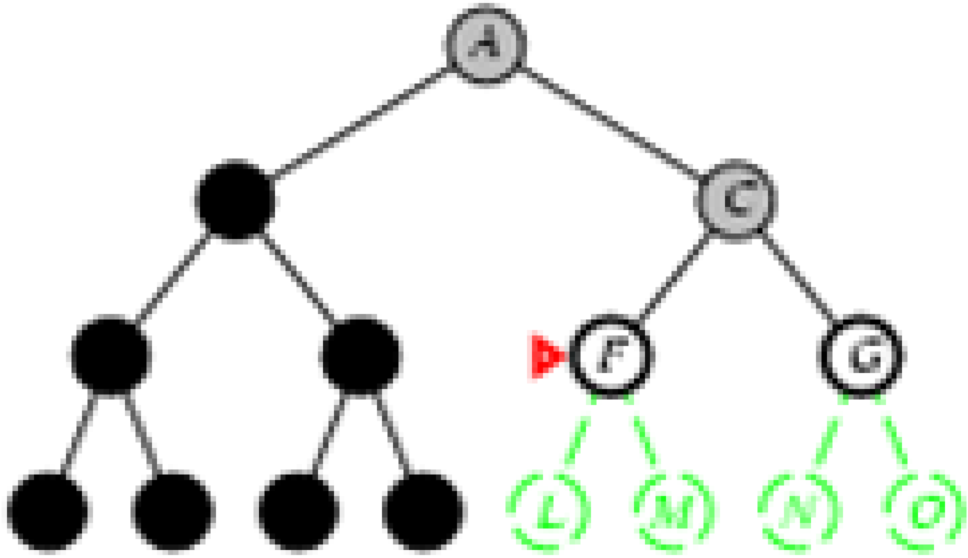


Image Credit: Artificial Intelligence A Modern Approach Second Edition by Stuart J. Russell and Peter Norvig by 2010.

Blind Search

3. Depth-first search (DFS)

- Expand deepest unexpanded node
- Nodes are stored in LIFO stack (put successors at front)

Stack: [L,M,G]

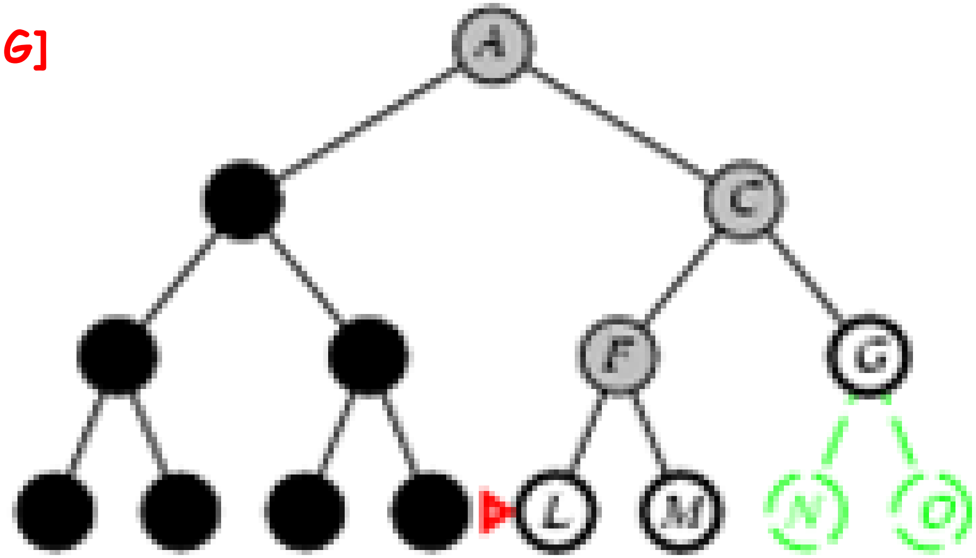


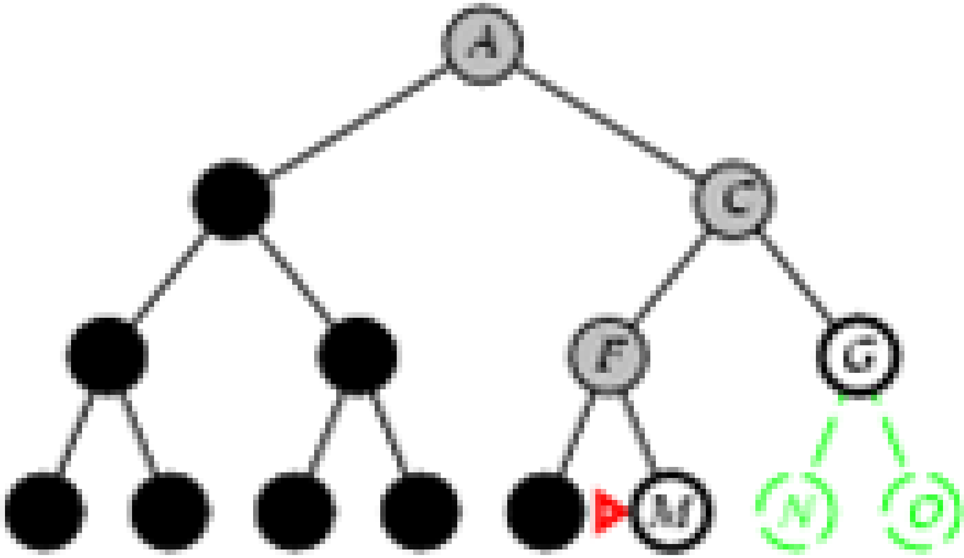
Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

3. Depth-first search (DFS)

- Expand deepest unexpanded node
- Nodes are stored in LIFO stack(put successors at front)

Stack: [M,G]



Goal is found!

Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search, evaluation

3. Depth-first search(DFS)

- Complete? No (fails in infinite-depth spaces, spaces with loops)
- Time? $O(b^m)$
- Space? $O(b.m)$
- Optimal? No

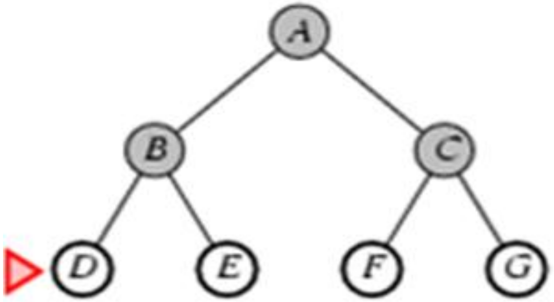


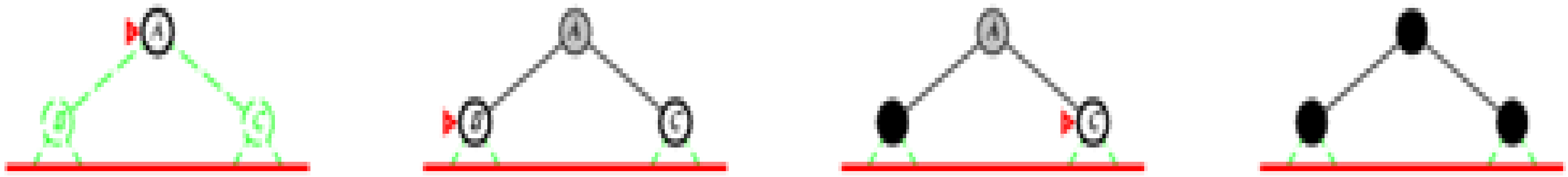
Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

4. Depth-limit search (DLS)

- Expand deepest unexpanded node until reach limit L
- Equivalent to depth-first search with depth limit L

❖ **Ex**: Let L=1



Goal is not found !

Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind search, evaluation

4. Depth-limit search(DLS)

- Complete? No (if d > L)
- Time? $O(b^l)$
- Space? $O(b.l)$
- Optimal? No

d: goal depth
L: depth Limit value

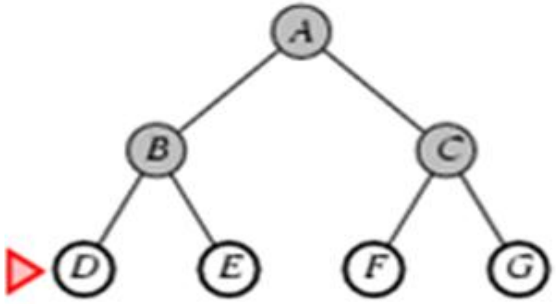


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

5. Iterative Depth-search (IDS)

- Expand deepest unexpanded start with L=0
- Repeated implementation of DFS with different L

Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

5. Iterative Depth-search (IDS)

- Expand deepest unexpanded start with L=0
- Repeated implementation of DFS with different L

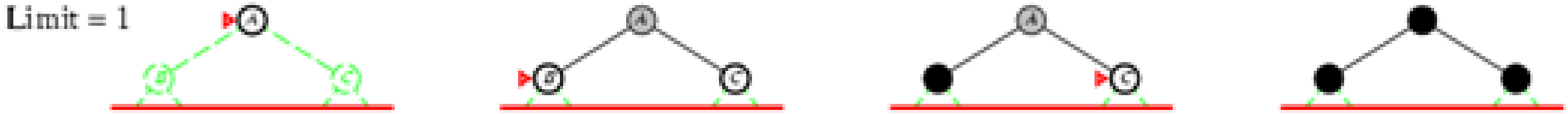


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

5. Iterative Depth-search (IDS)

- Expand deepest unexpanded start with L=0
- Repeated implementation of DFS with different L

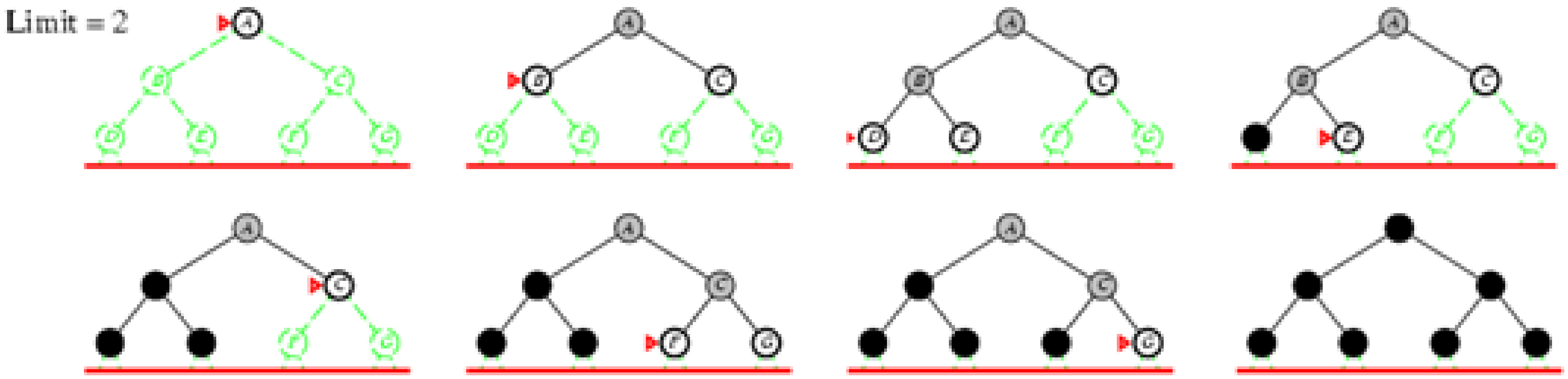
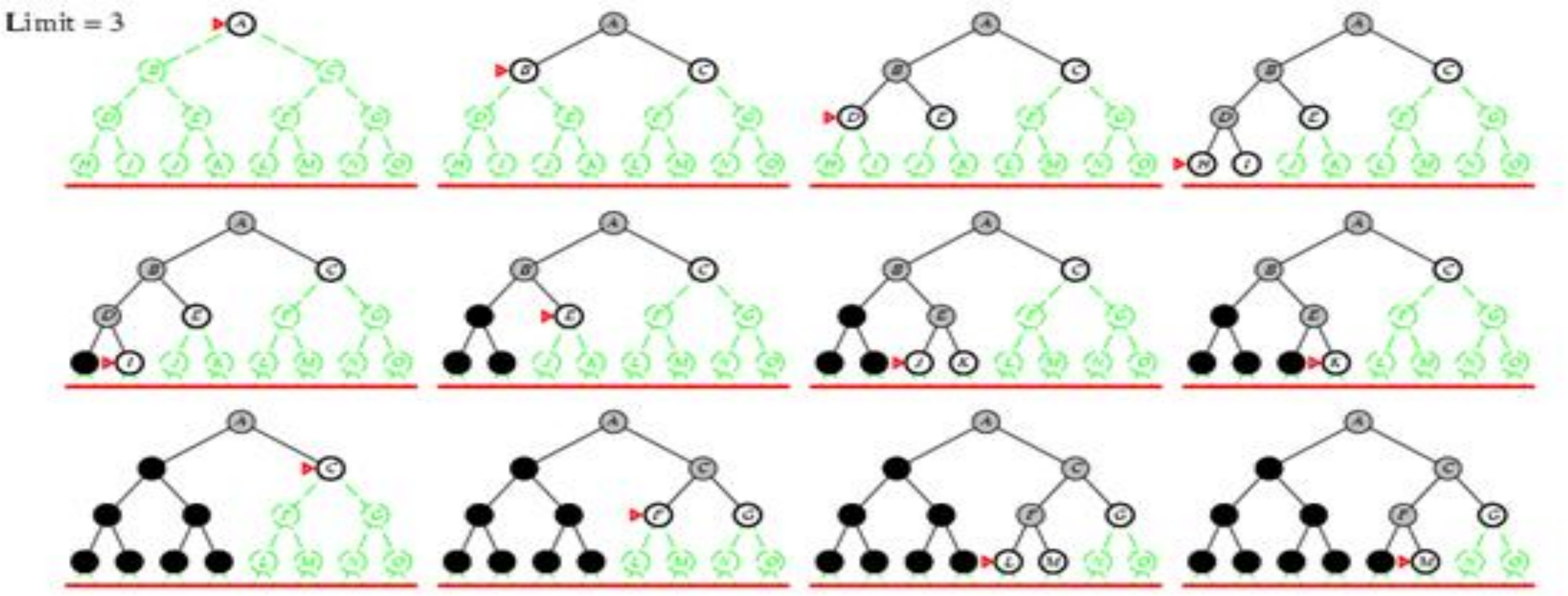


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search

5. Iterative Depth-search (IDS)

- Expand deepest unexpanded start with L=0
- Repeated implementation of DFS with different L



Goal is found !

Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search, evaluation

5. Iterative Depth-search (IDS)

Complete?	Yes
Time?	$O(b^d)$
Space	$O(b.d)$
Optimal?	Yes (if all trans. have same cost)

Blind Search

6. Bidirectional search (BS)

- BFS search simultaneously forward from START and backward from GOAL
- Solution is found if the two searches meet

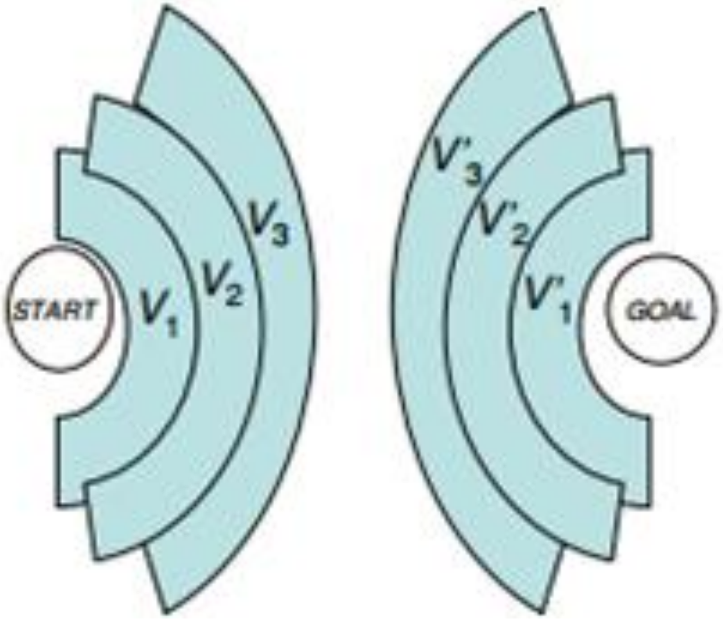


Image Credit: Artificial Intelligence A Modern Approach *Second Edition by Stuart J. Russell and Peter Norvig by 2010.*

Blind Search, evaluation

6. Bidirectional search (BS)

- Complete? Yes (if b is finite)
- Time? $O(b^{d/2})$
- Space? $O(b^{d/2})$
- Optimal? Yes (if all trans. have same cost)

Applications

- GPS Navigation systems: Google Maps, which can give directions to reach from one place to another using BFS

- Computer Networks: Peer to peer (P2P) applications such as the torrent clients need to locate a file that the client is requesting by applying BFS on the hosts on a network

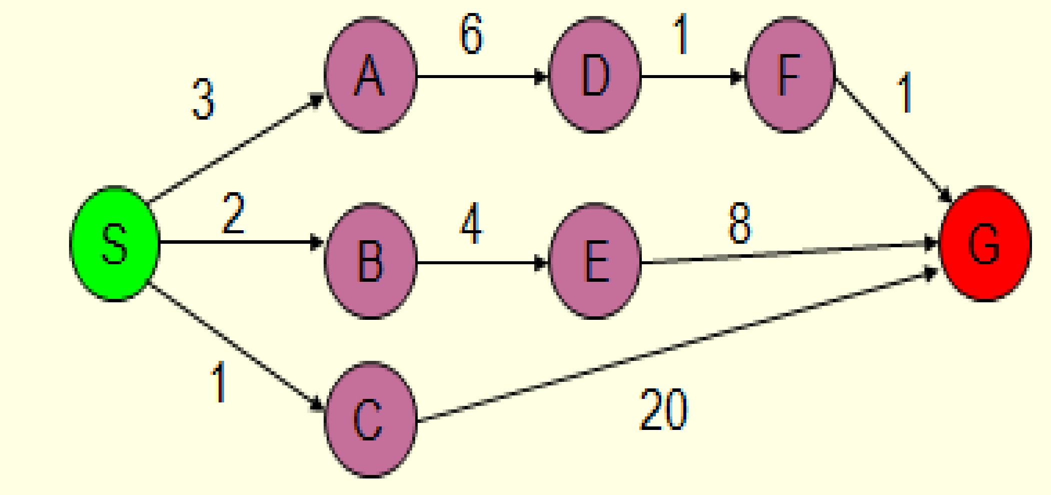
- Wireless Technology: in order to demand high data rates in wireless technology, MIMO (Multi input Multi output) employs BFS to find the shortest (Euclidian distances)

- Social Networks: Facebook treats each user profile as a node on the graph search space and two nodes are said to be connected if they are each other's friends

Assignment_1

What is the different between Tree search and Graph search ?

Assignment_2



based on this graph, initial state is S and goal state is G

1. Apply **BFS** to reach destination state
2. Apply **DFS** to reach destination state
3. Apply **DLS** to reach destination state (depth limit 2)
4. Apply **UCS** to reach destination state , is this the optimal solution? Why?
5. For each algorithm:

   - Compute solution cost, (if there is a solution)
   - Express time and space in terms of # nodes

Thank You !