Artificial Intelligence

By Dr. Naglaa fathy
Lecturer, Computer Science Department
Faculty of Computers and Artificial Intelligence
Benha University

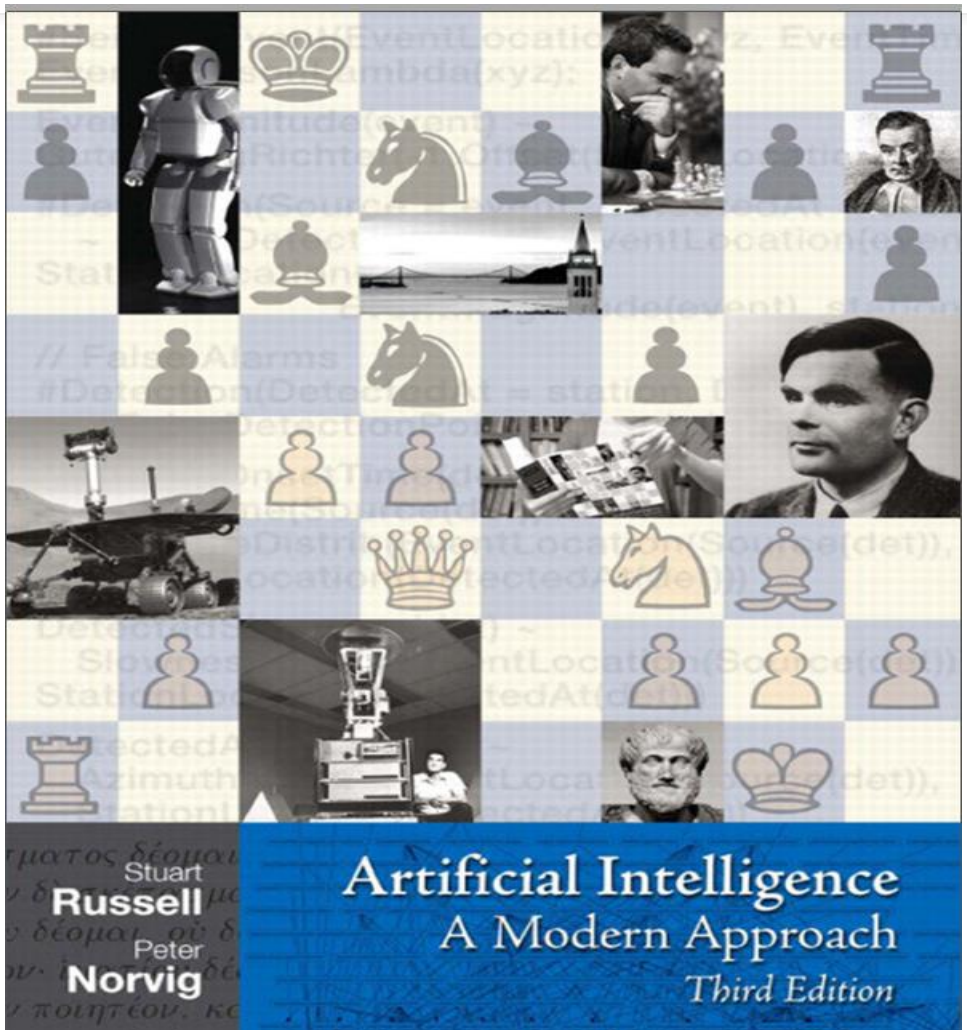
Lecture 4
Solving problem by searching

Lectures References

Artificial Intelligence
A Modern Approach

Third Edition

Stuart J. Russell and Peter Norvig

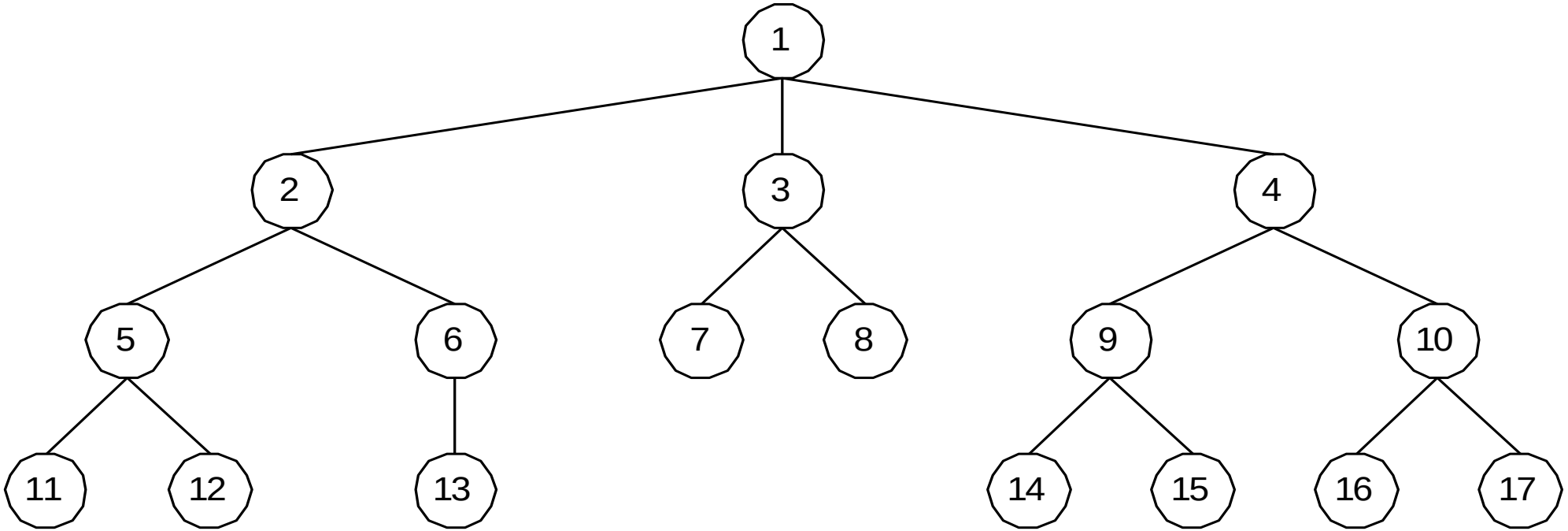


Uninformed search strategies

- Breadth-first search
- Uniform-cost search
- Depth-first search
- Depth-limited search
- Iterative deepening(depth) search

Breadth-first search

- Expand shallowest unexpanded node



Uniform-cost search

- Uniform-cost Search: Expand node with smallest path cost g(n).
- Implementation: *fringe* = queue ordered by path cost
- Nodes are sorted in ordered queue.
- Equivalent to breadth-first if all step costs all equal.

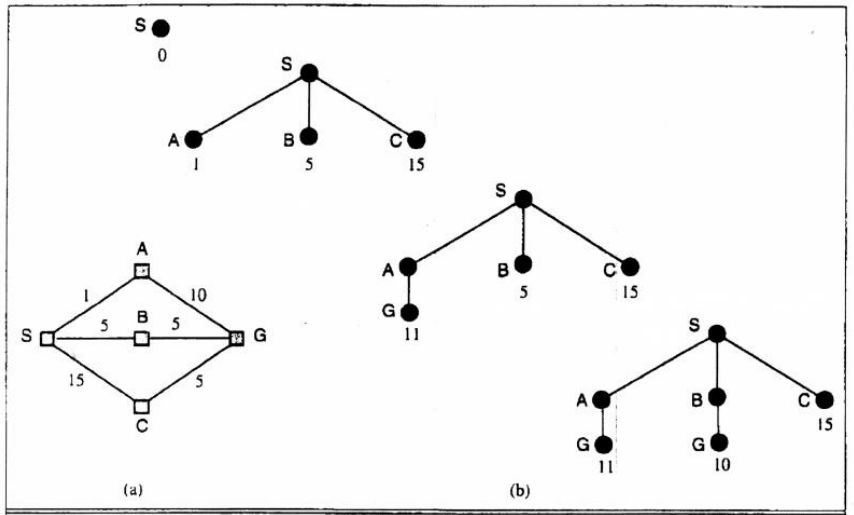
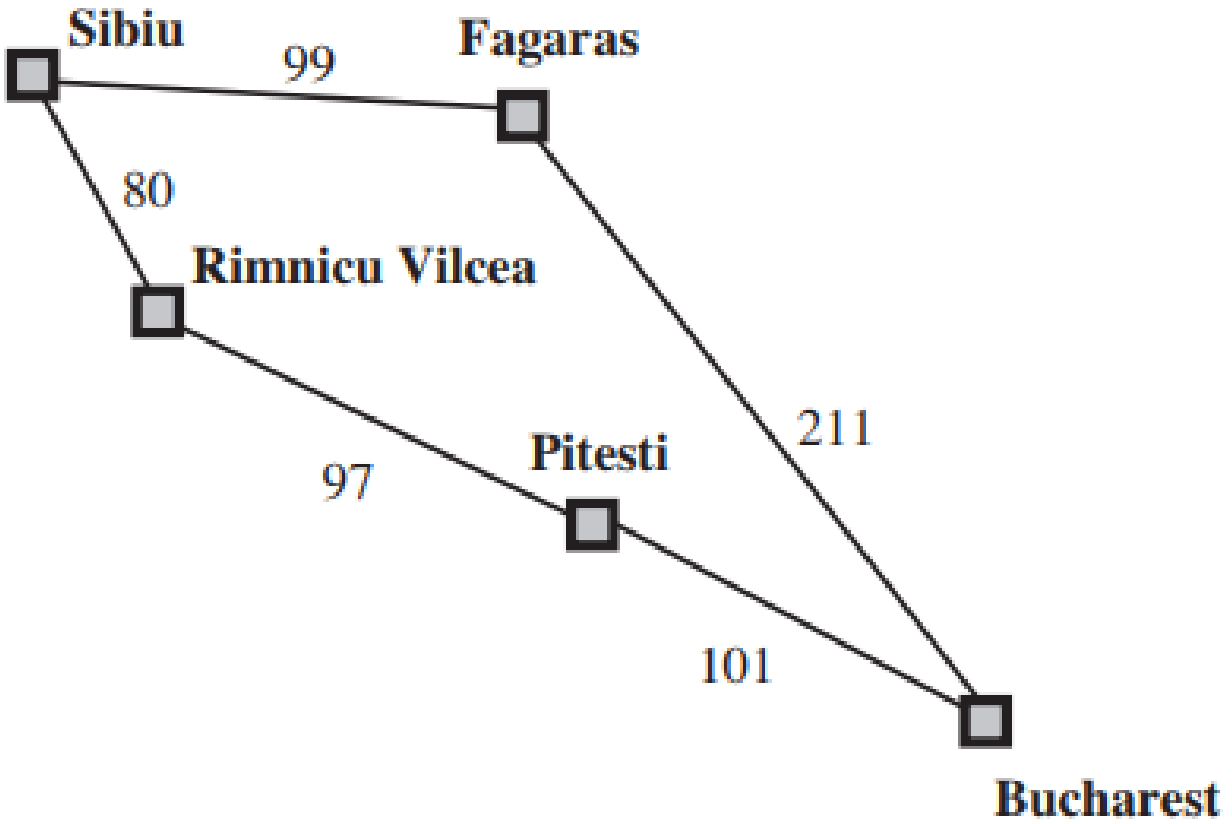


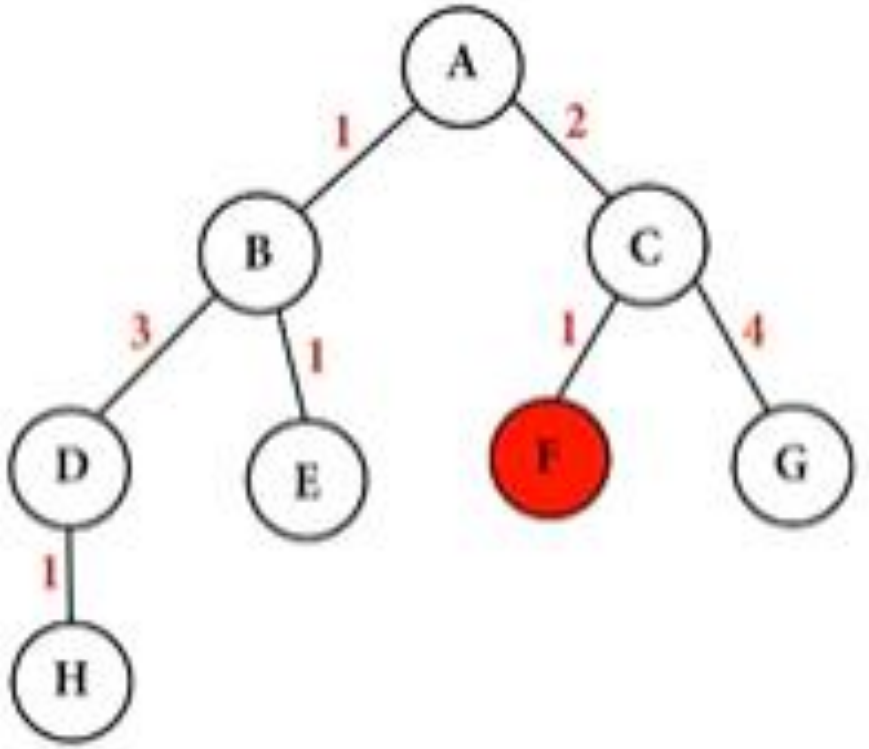
Figure 3.13 A route-finding problem. (a) The state space, showing the cost for each operator. (b) Progression of the search. Each node is labelled with $g(n)$. At the next step, the goal node with $g = 10$ will be selected.

Example

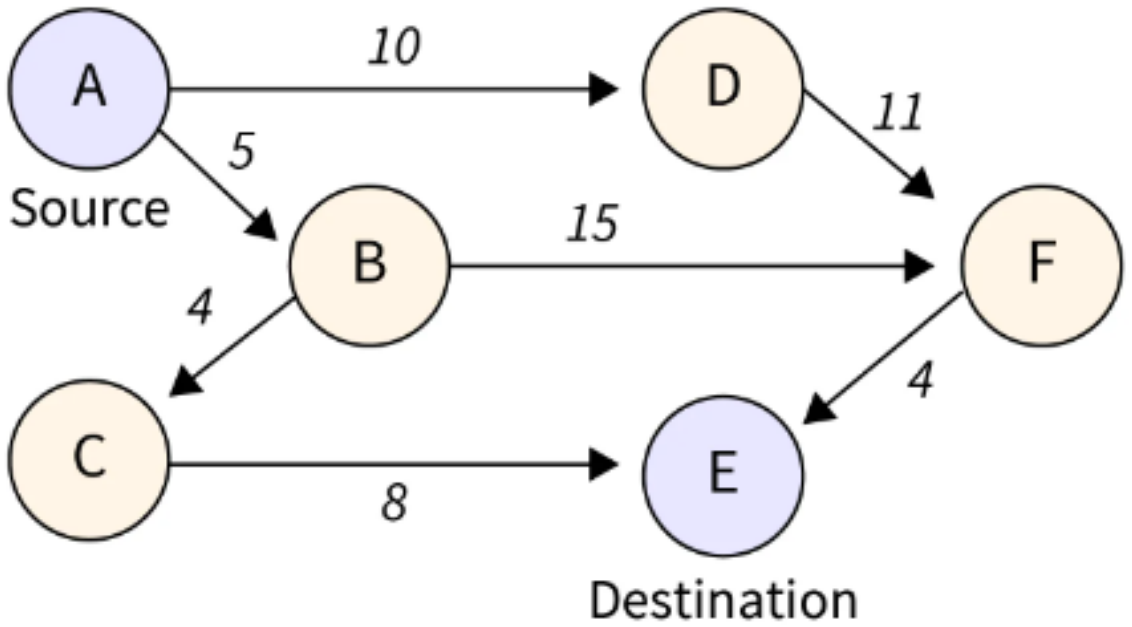

Sibiu
99
Fagaras
80
Rimnicu Vilcea
211
Pitesti
97
101
Bucharest

Example

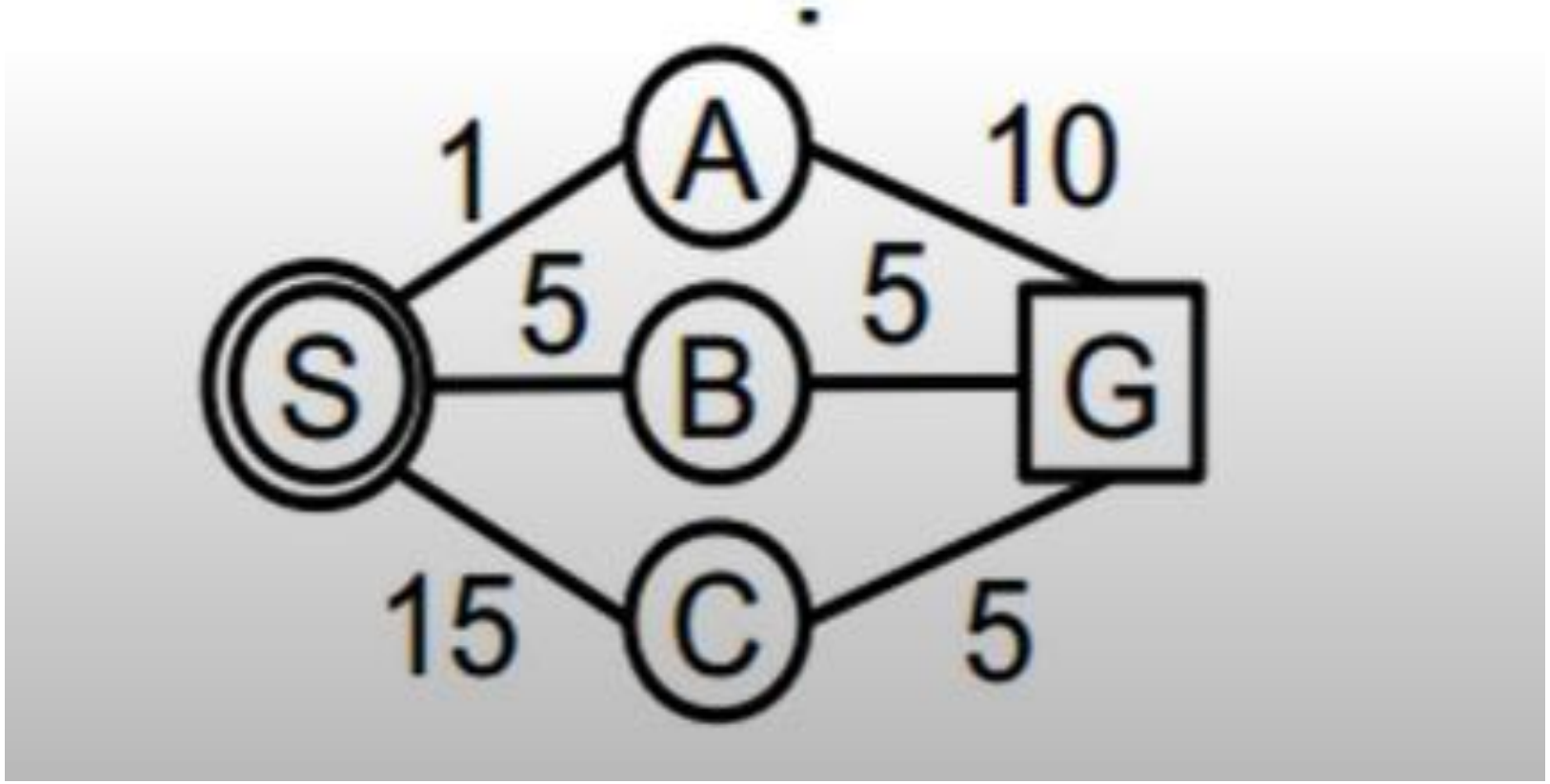
Uniform Cost Search
(UCS)



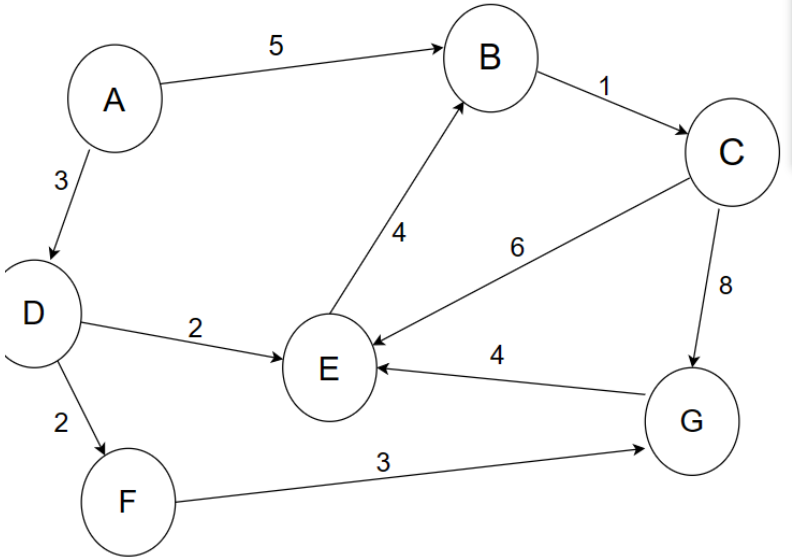
Example



Example

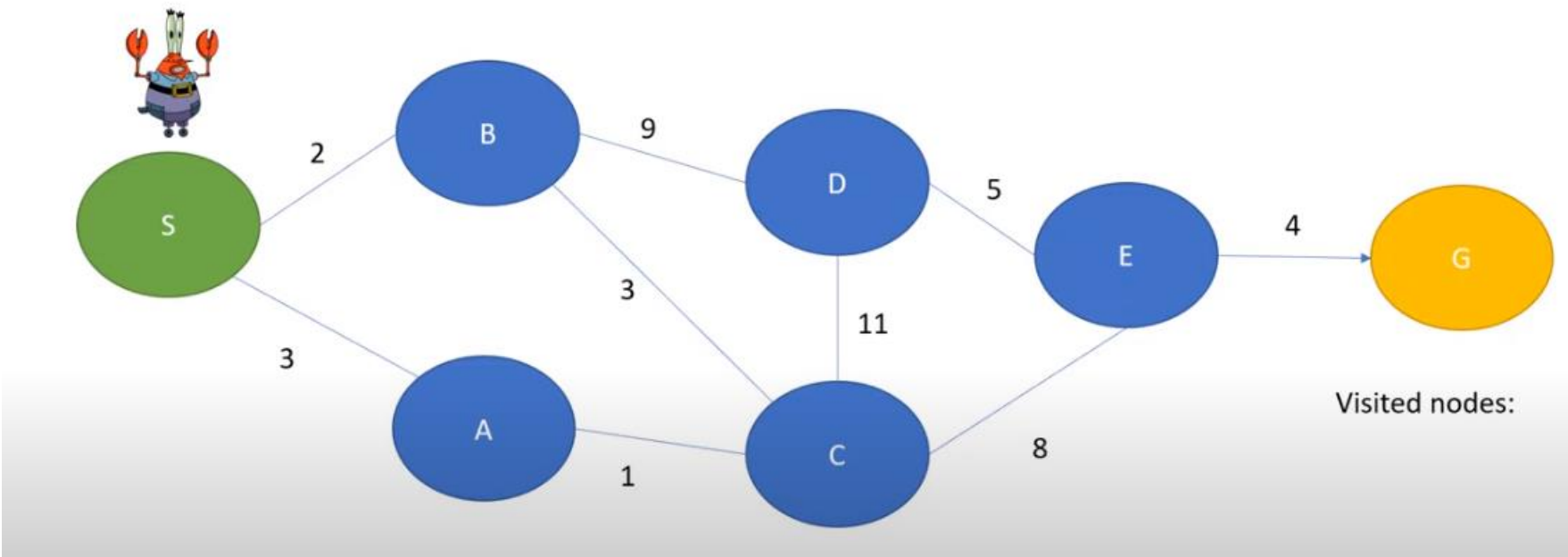


Example

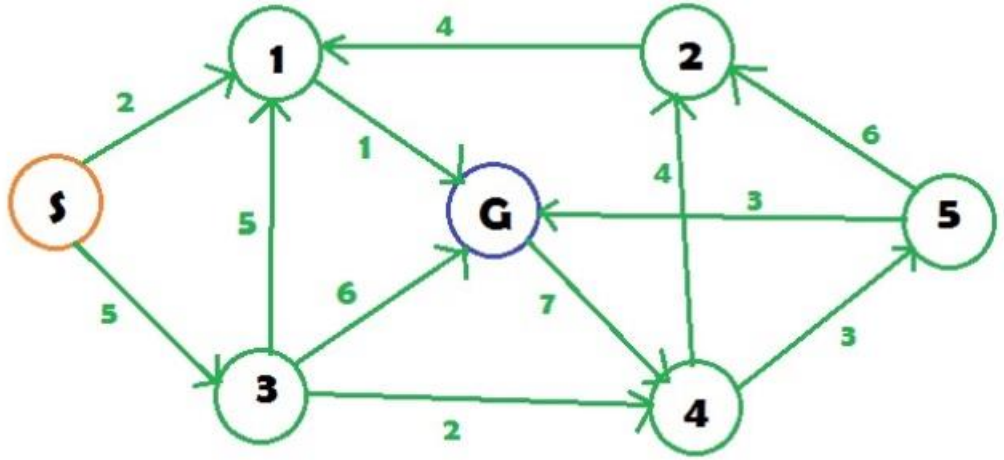


	Frontier List	Expand List	Explored List
1.	{A,0)}	A	NULL
2.	{(A-**D**, 3), (A-B, 5)}	D	{A}
3.	{(A-**B**, 5), (A-D-E, 5), (A-D-F, 5)}	B	{A, D}
4.	{(A-D-**E**, 5), (A-D-F, 5), (A-B-C, 6)}	E	{A, D, B}
5.	{(A-D-**F**, 5), (A-B-C, 6), ~~(A-D-E-B, 9)~~} *here **B** is **already explored**	F	{A, D, B, E}
6.	{(A-B-**C**, 6), (A-D-F-G,8)}	C	{A, D, B, E, F}
7.	{(A-D-F-**G**,8),~~(A-B-C-E,12)~~, (A-B-C-G, 14)} ***here **E** is **already explored**	G	{A, D, B, E, F, C}
8.	{(A-D-F-G,8)}	NULL	{A, D, B, E, F, C, **G**} # GOAL Found!

Example



Assignment



Properties of uniform cost search

Completeness: ? Yes, if step cost $\geq \varepsilon$

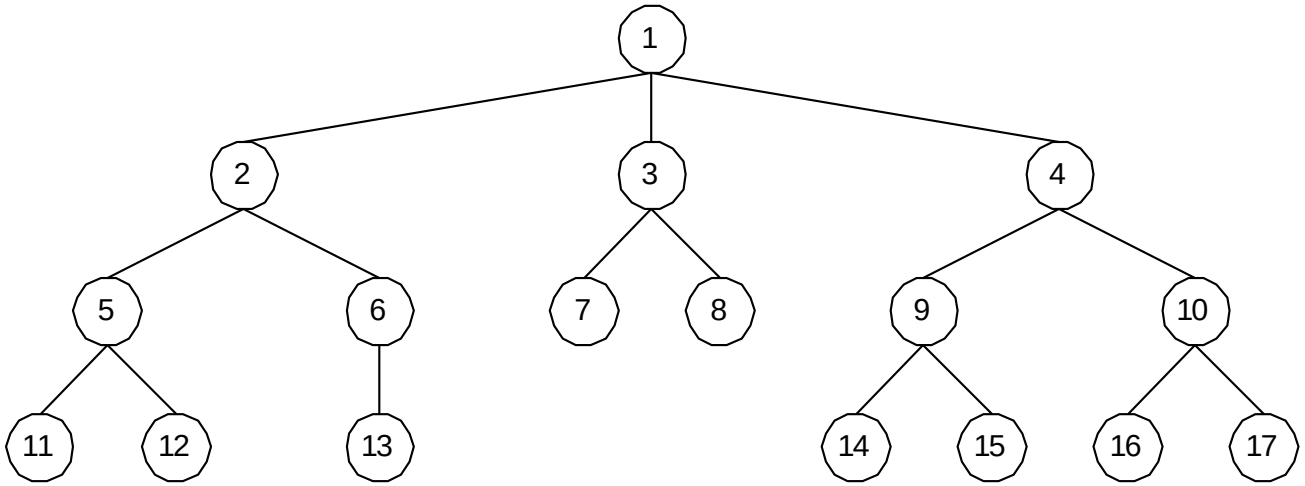
Optimality: Yes, for any step cost

Time complexity: # of nodes with path cost $\leq$ cost of optimal solution.

Space complexity: # of nodes on paths with path cost $\leq$ cost of optimal solution.

Depth-first search

Expand deepest unexpanded node

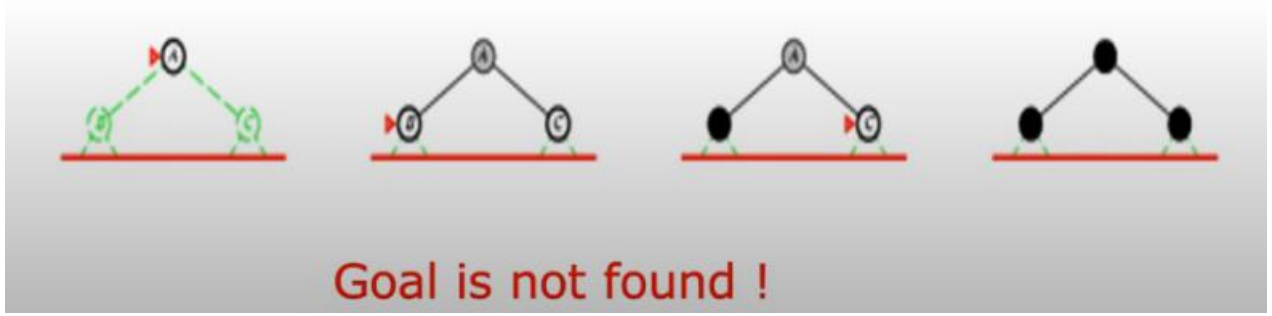


Depth-limited search

- The failure of depth-first search in infinite state spaces can be prevented by giving it a search limit of L.
- A variation of depth-first search that uses a depth limit
  - mitigate the problem of unbounded trees
  - Search to a predetermined depth L
  - Nodes at depth l have no successors
- Equal depth-first search with depth limit L
- Expand deepest unexpanded node until reaches limit L.

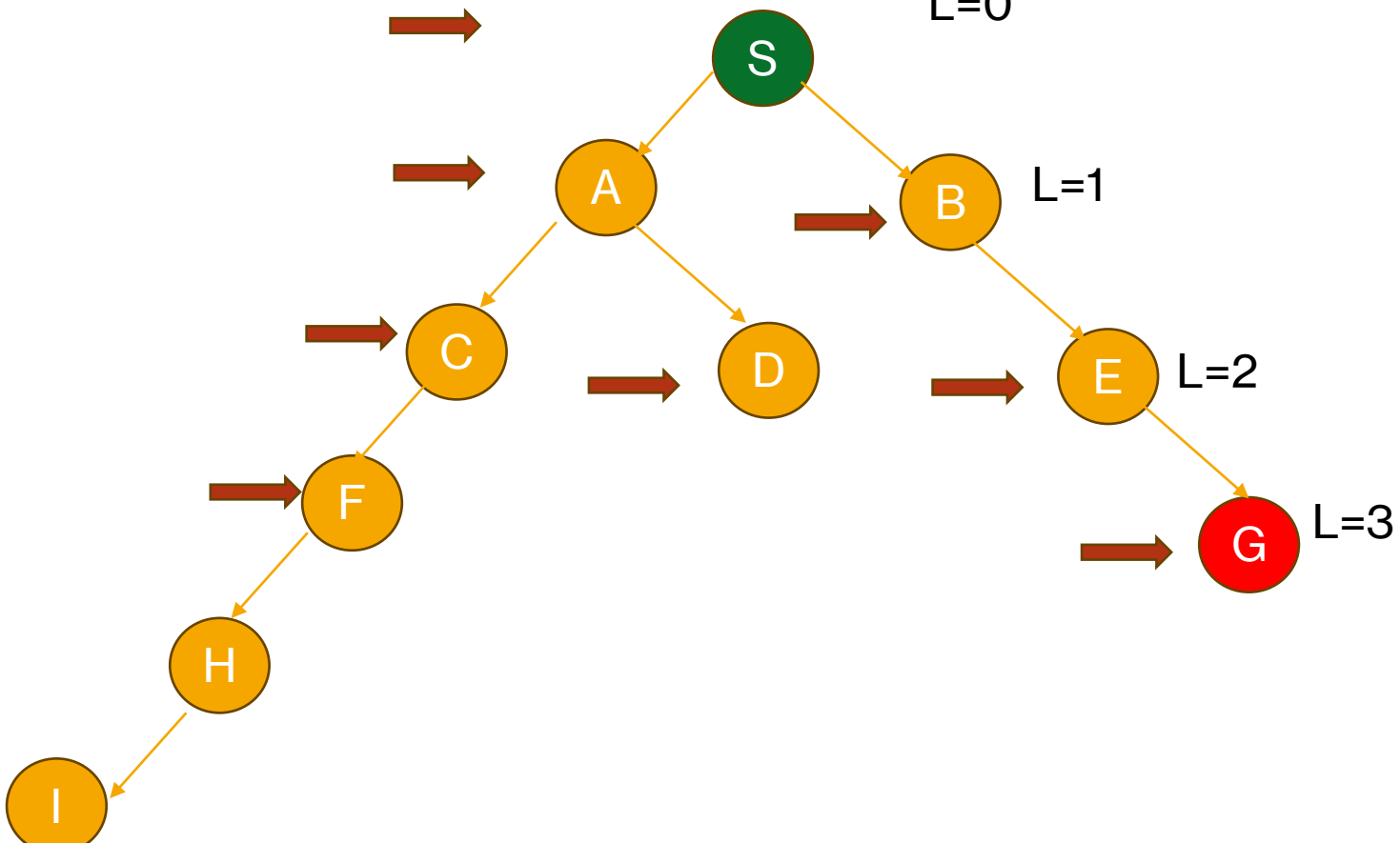
Example

Let L=1



Example

Given limit =3



Visited node
S
A
C
F
D
B
E
G

Properties of limited depth search

Completeness: ? No: if d>L

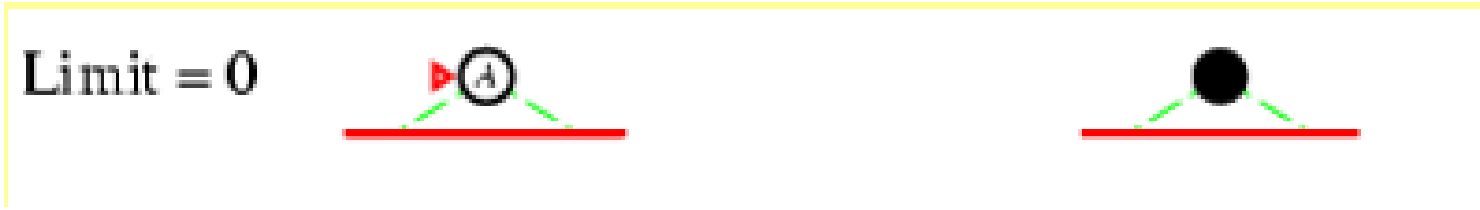
Optimality: No (if we choose L> d)

Time complexity: $O(b^l)$

Space complexity: $O(bl)$

Iterative deepening depth-first search

- Expand deepest unexpanded node starting with L=0.
- Repeated implementation of DLS with different L.

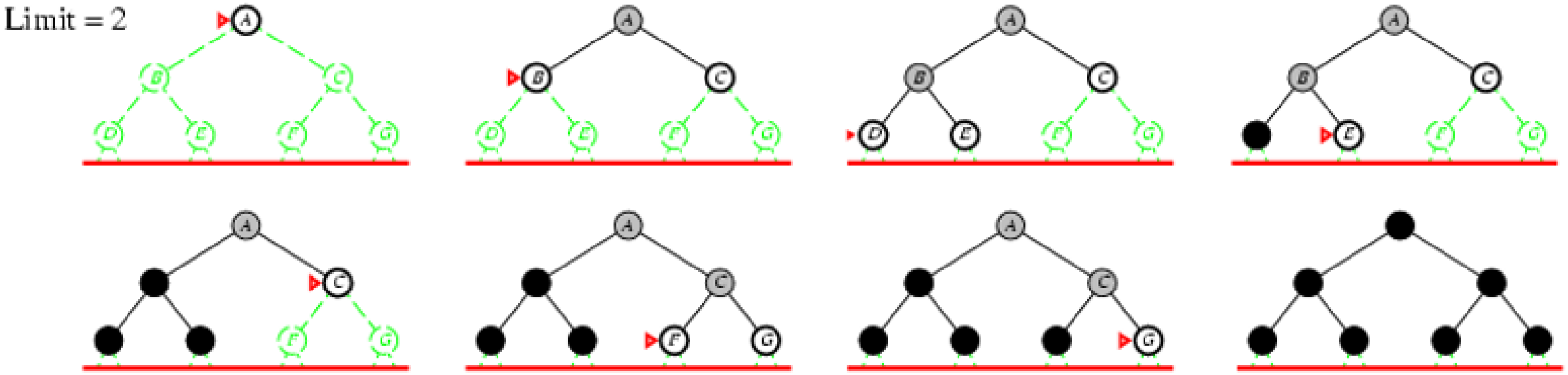


Iterative deepening search *L*=1

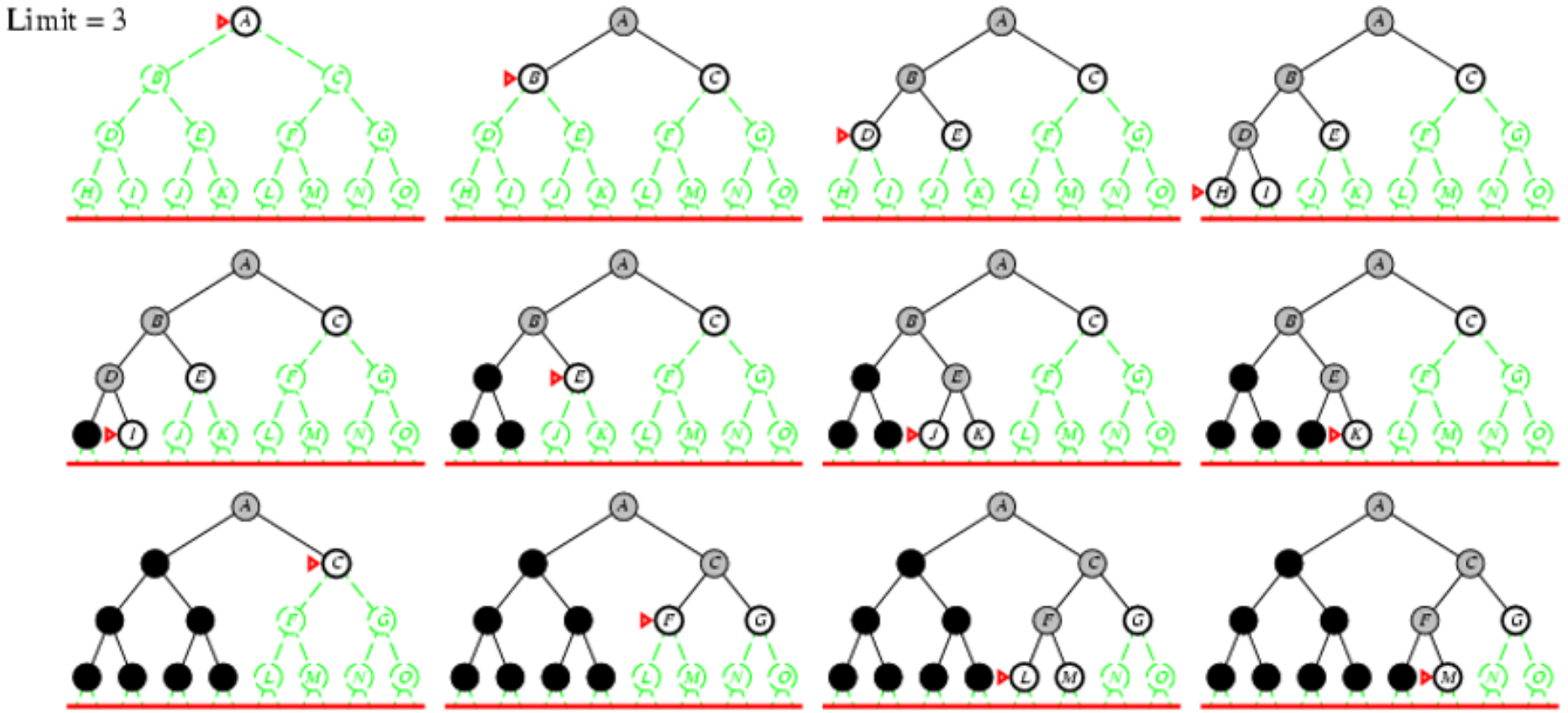
Iterative deepening search *L*=1

Limit = 1

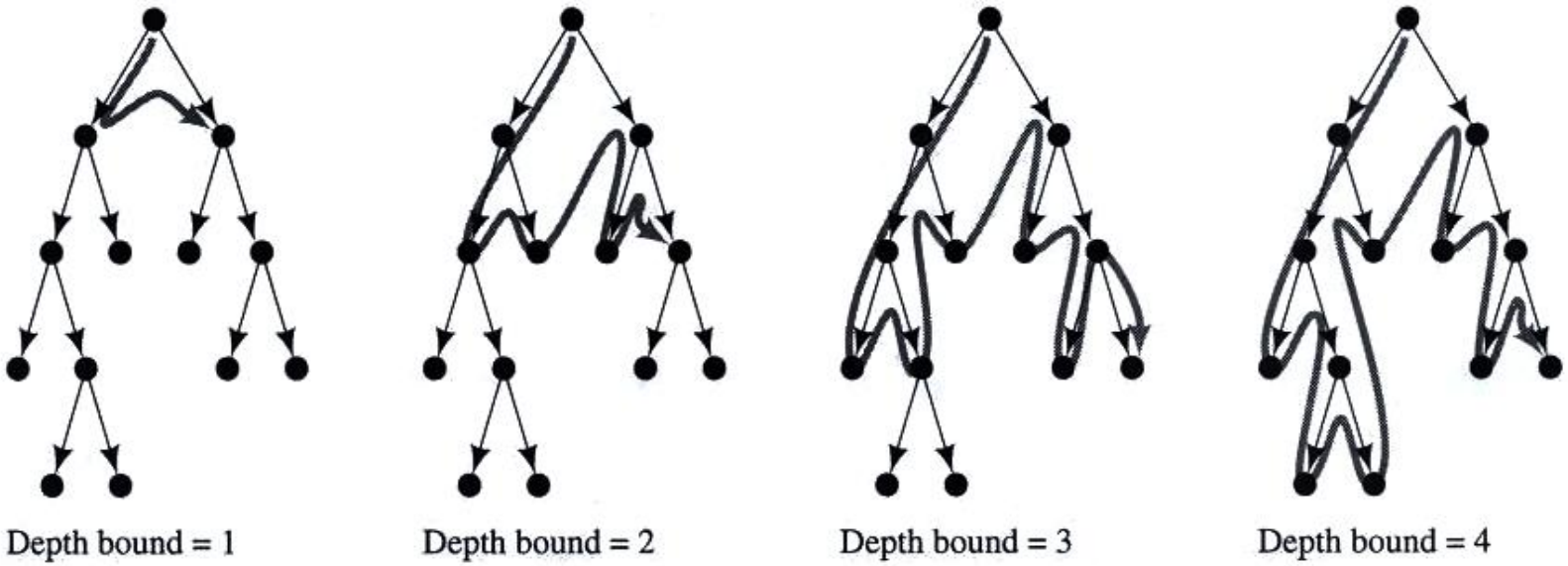
Iterative deepening search *L*=2



Iterative deepening search *L*=3

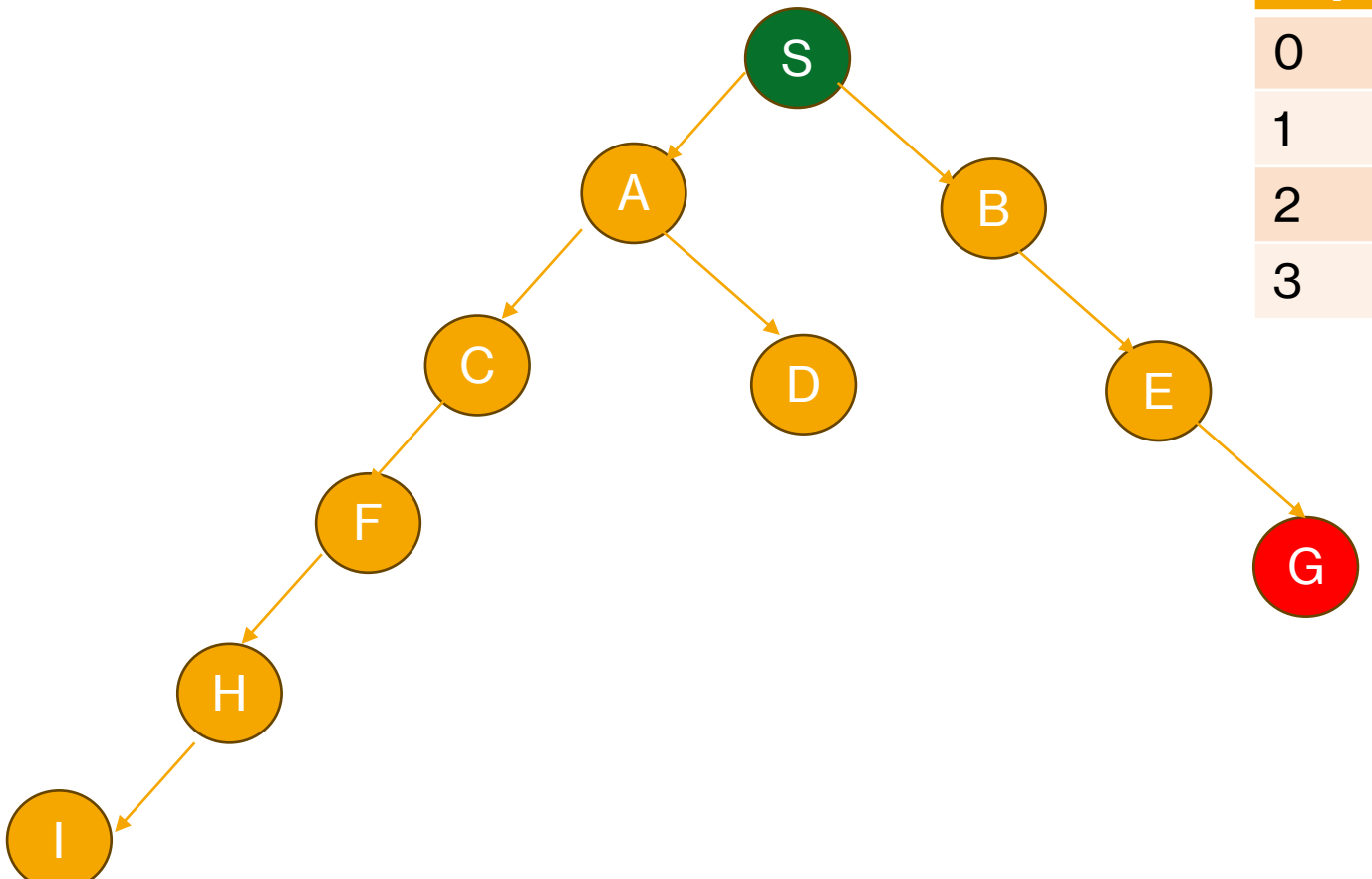


Example IDS



Stages in Iterative-Deepening Search

Example



depth	Visited node
0	s
1	S A B
2	S A C D B
3	S A C F D B E G

Properties of Iterative deepening search

Completeness: ? Yes

Optimality: yes

Time complexity: $O(b^d)$

Space complexity: $O(bd)$


AI
THANKYOU