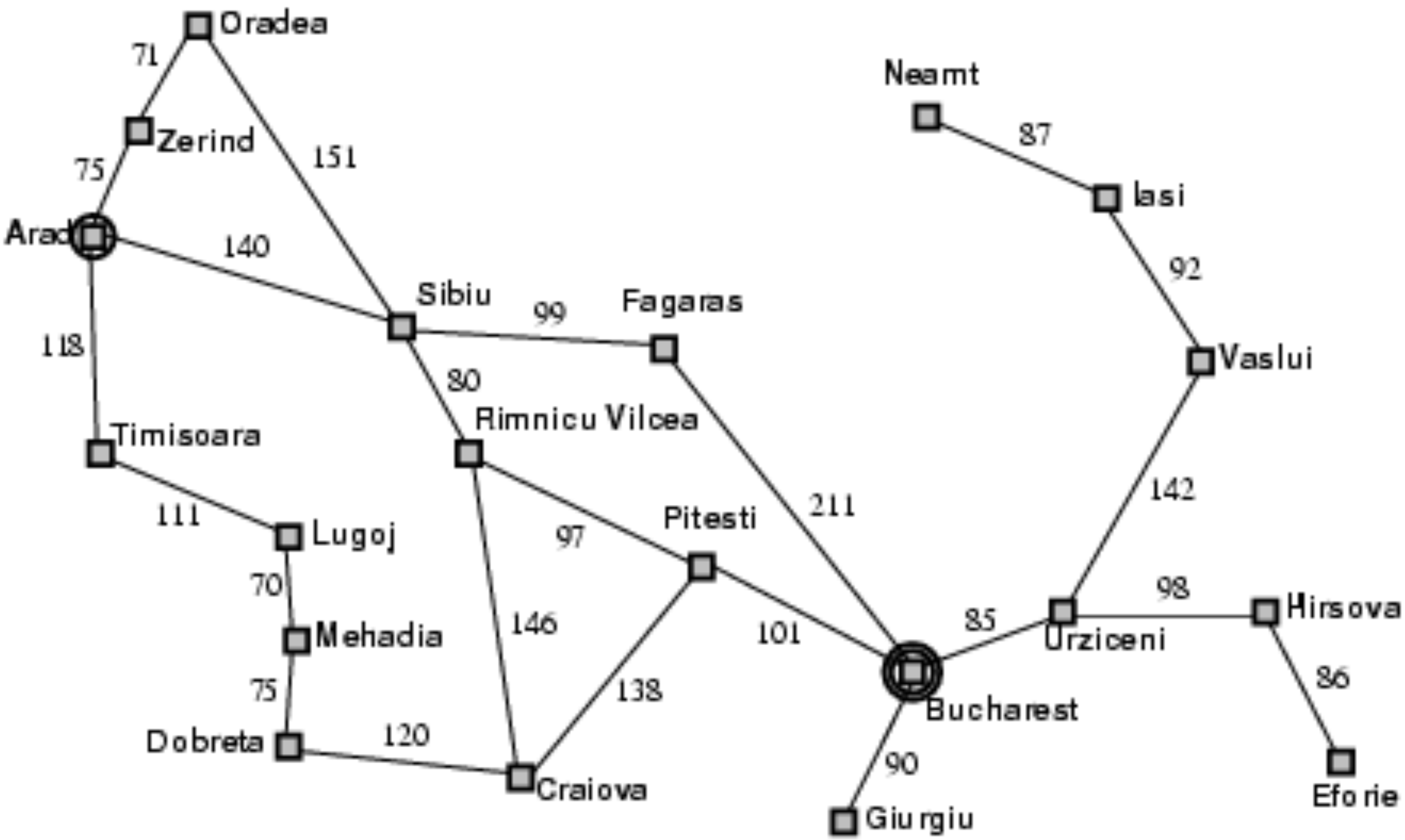
Lecture 6: Depth-first search

Artificial Intelligence
CS-GY-6613-I

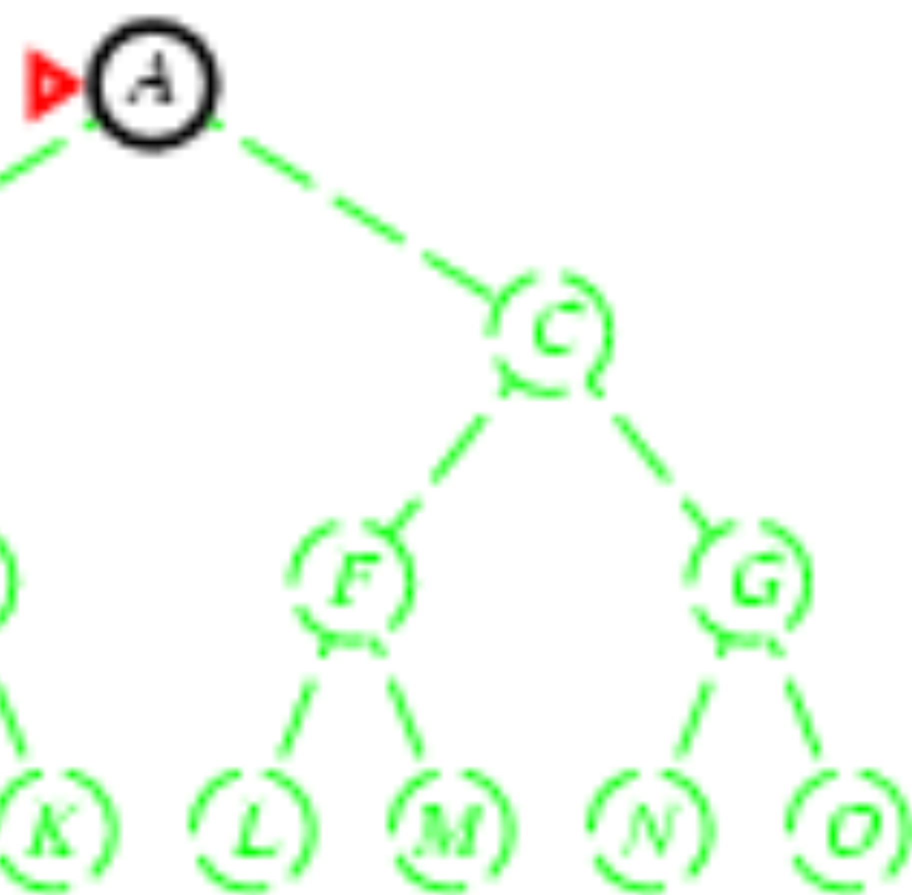
Julian Togelius
julian.togelius@nyu.edu

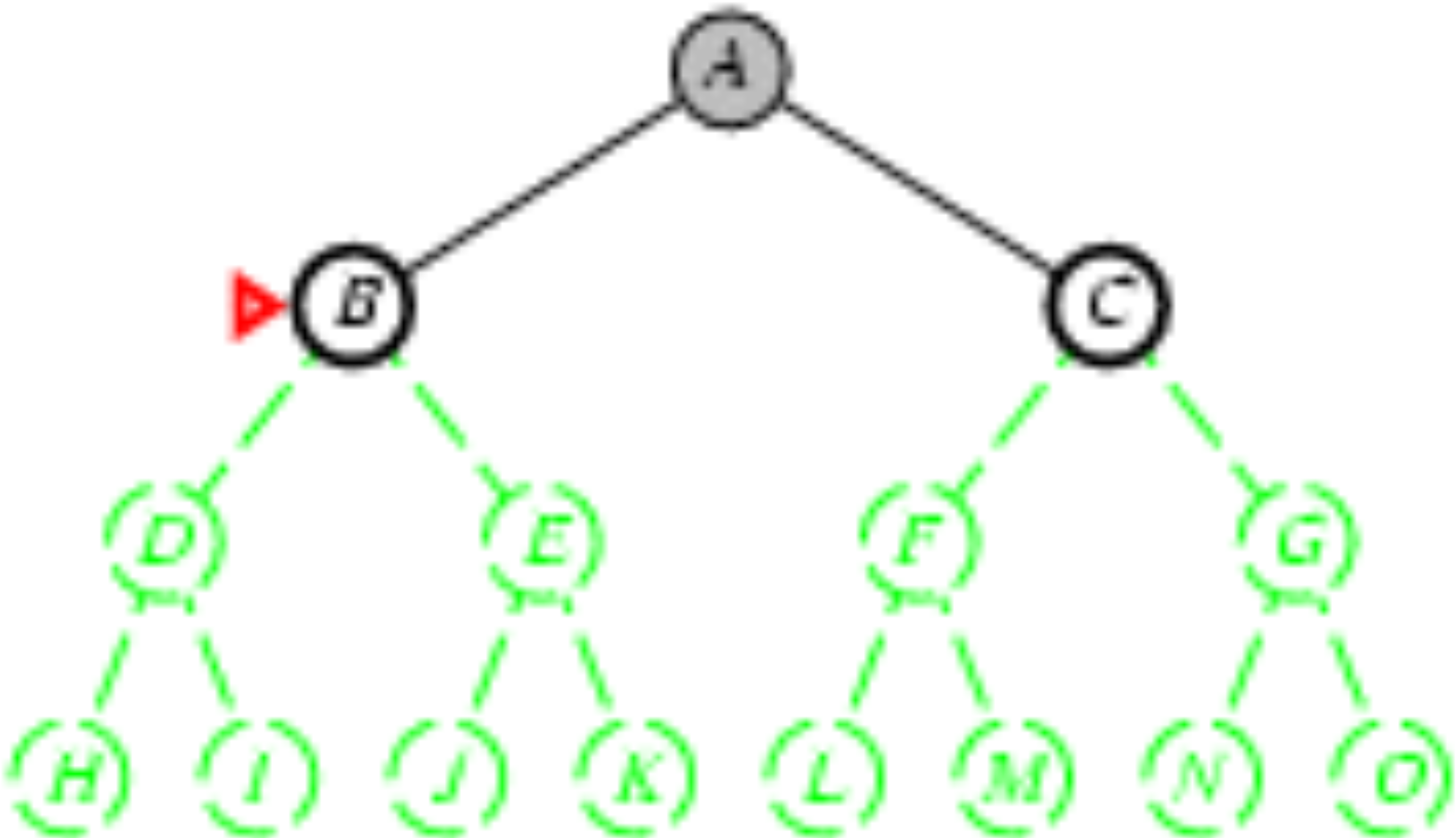
From Arad to Bucharest

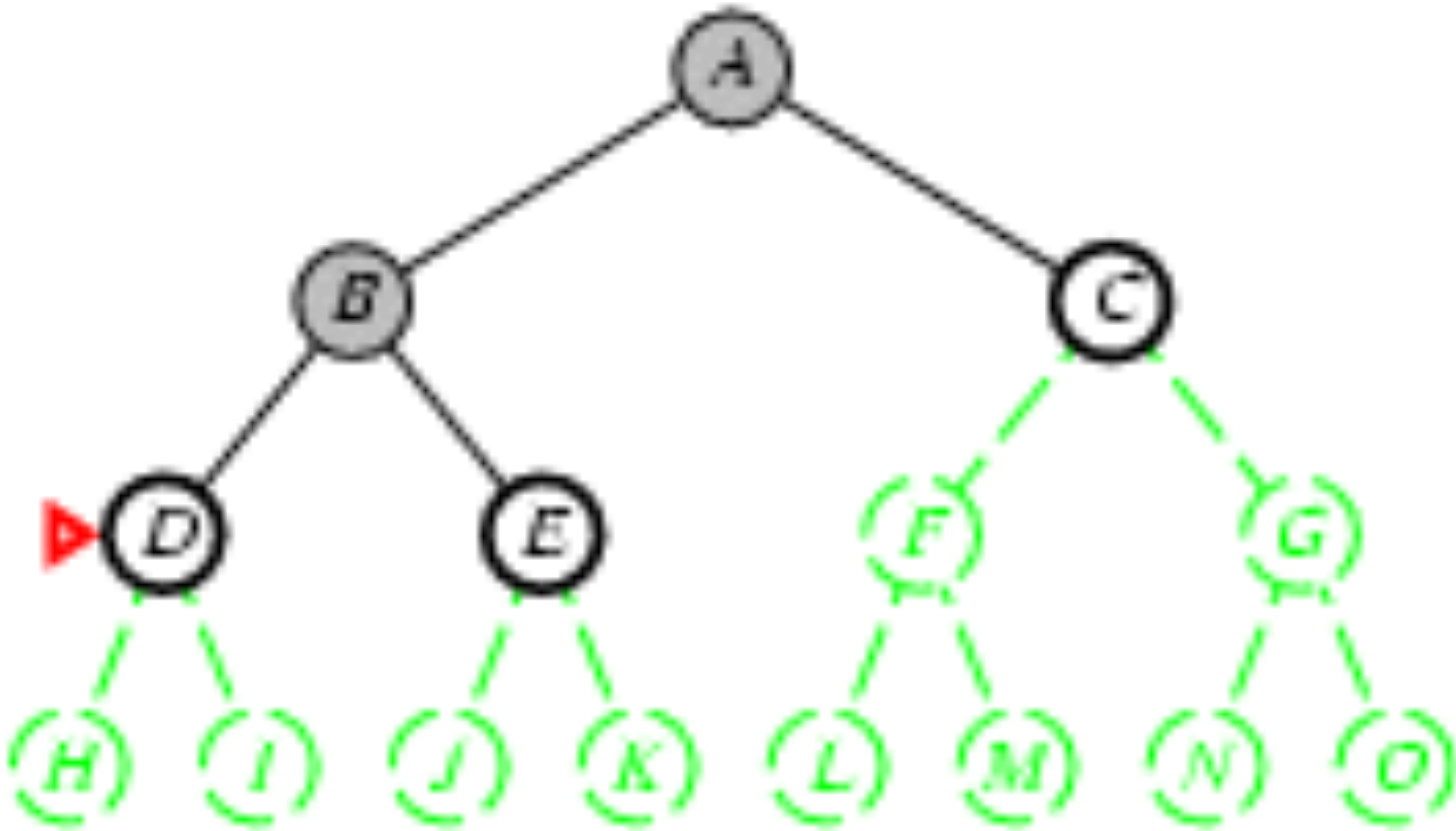


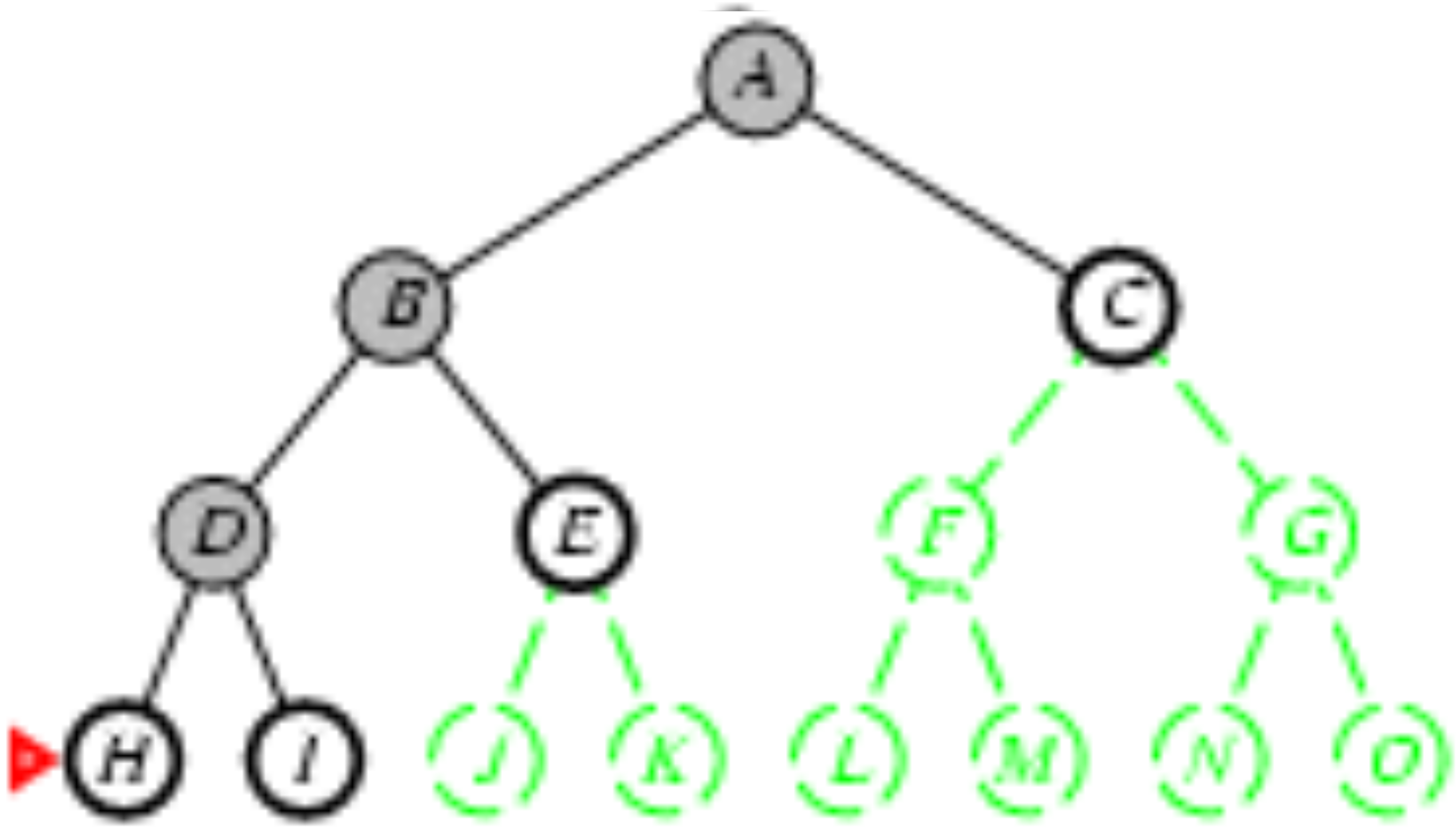
Depth-first search

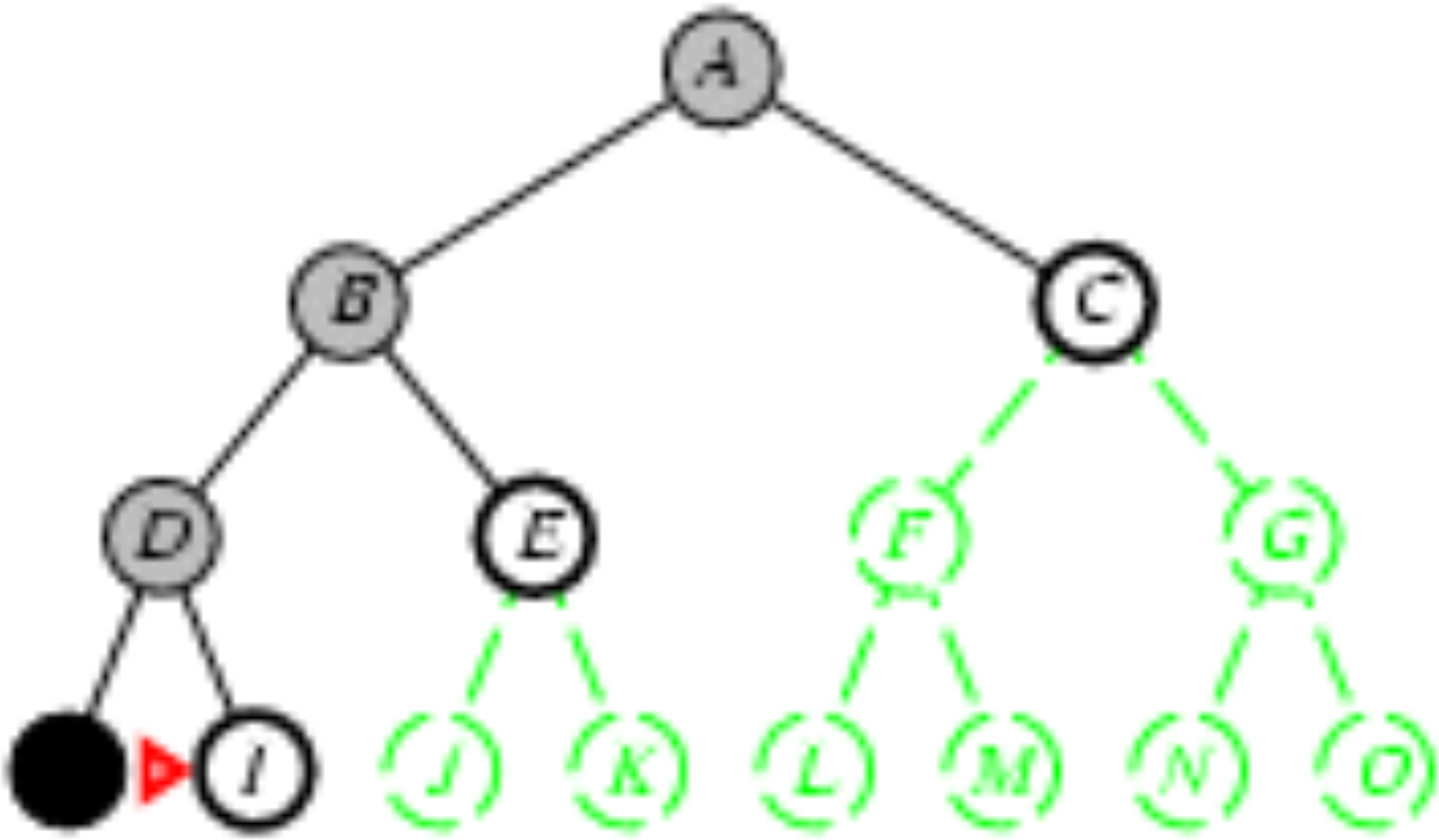
- Expand deepest unexpanded node
- Implementation: fringe = LIFO queue, i.e., put successors at front

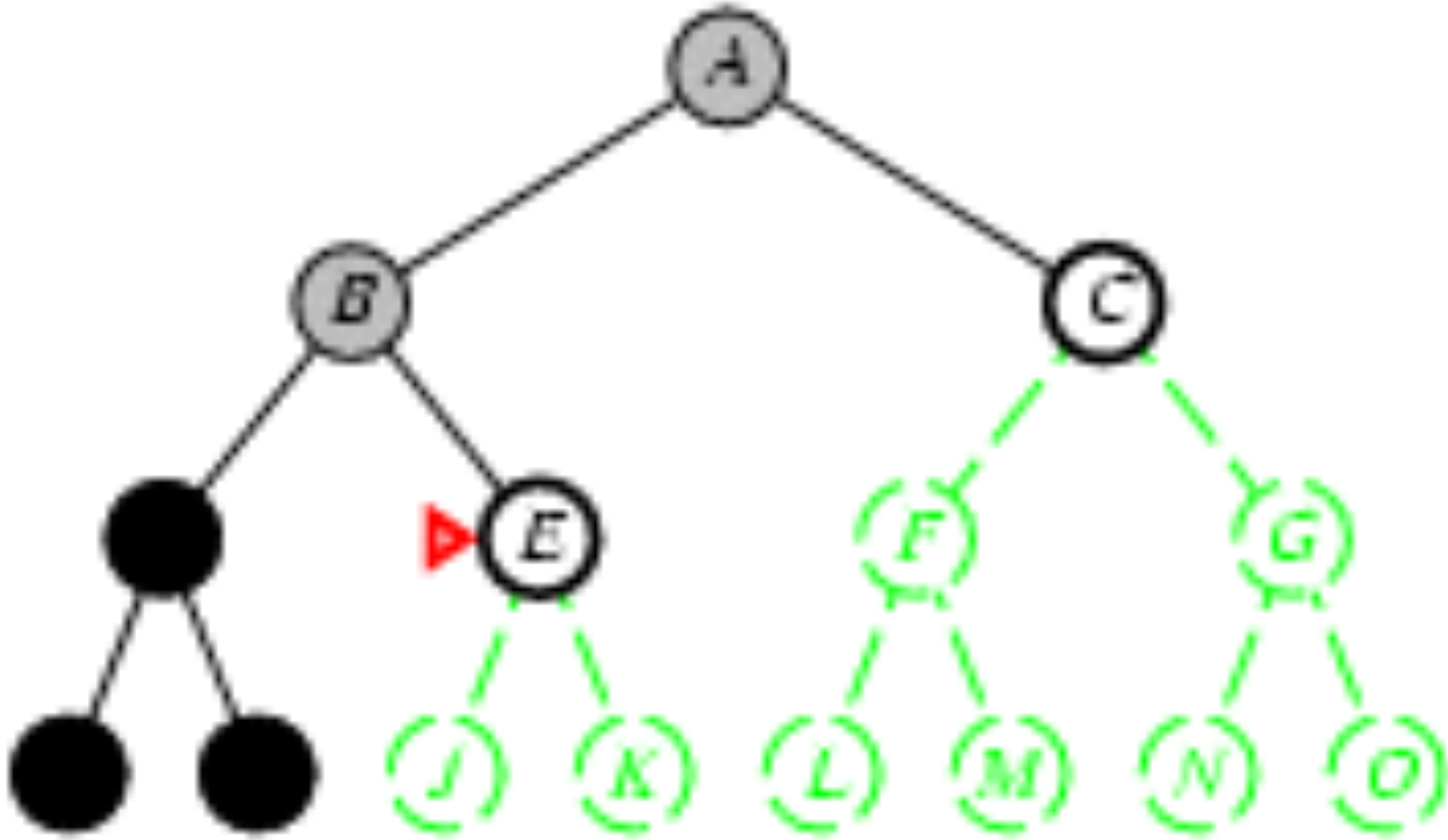


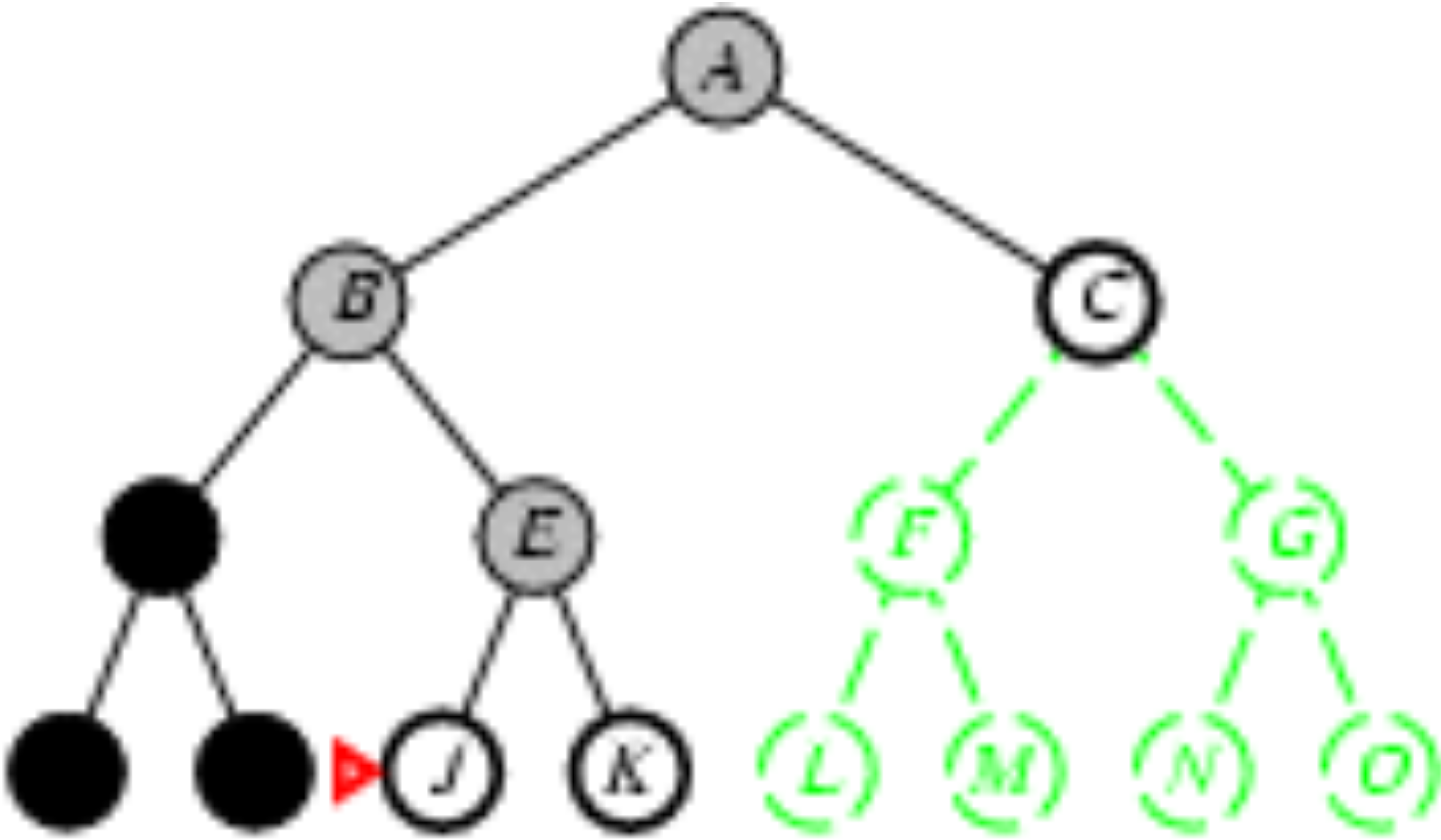

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O

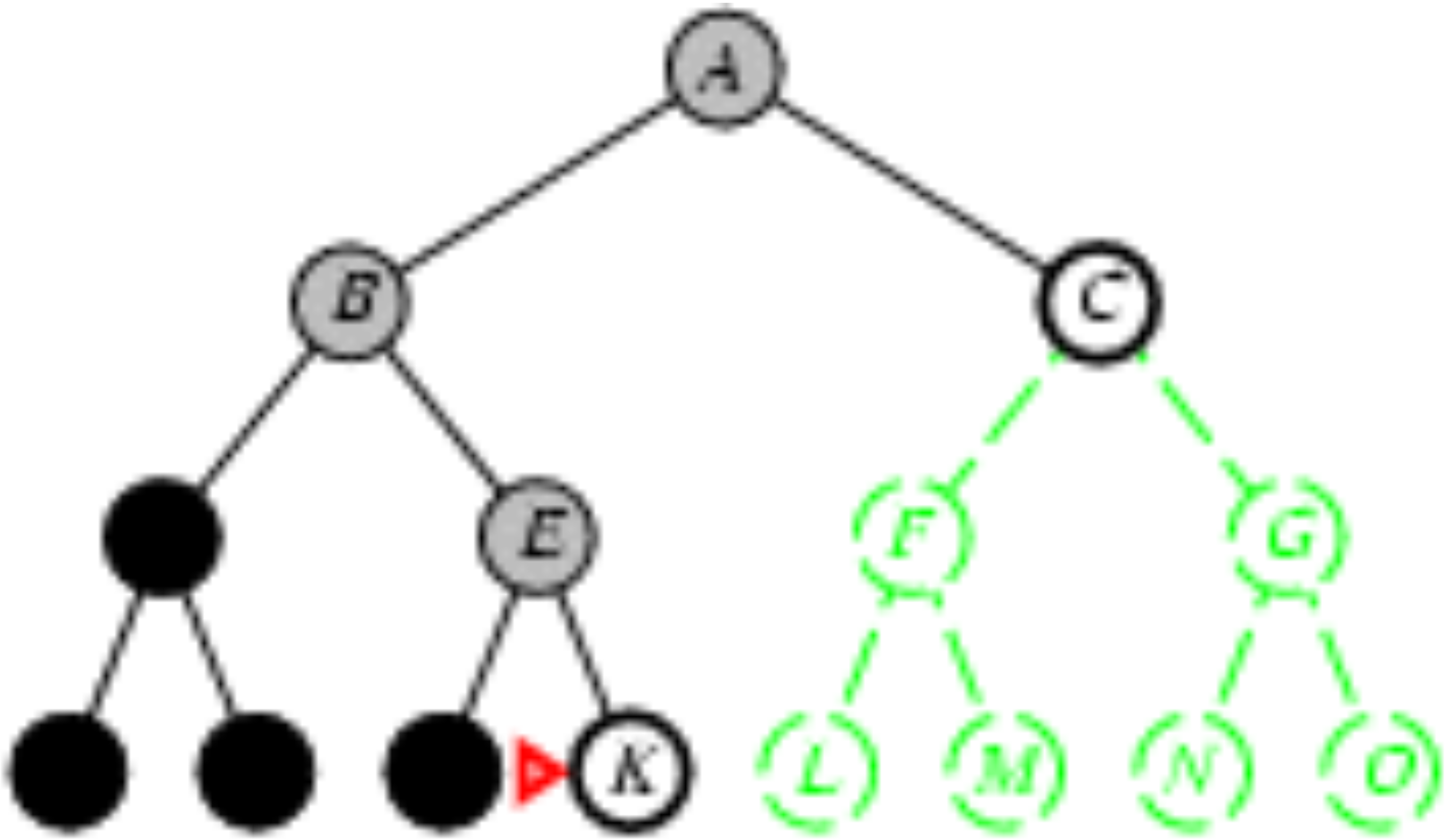

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O

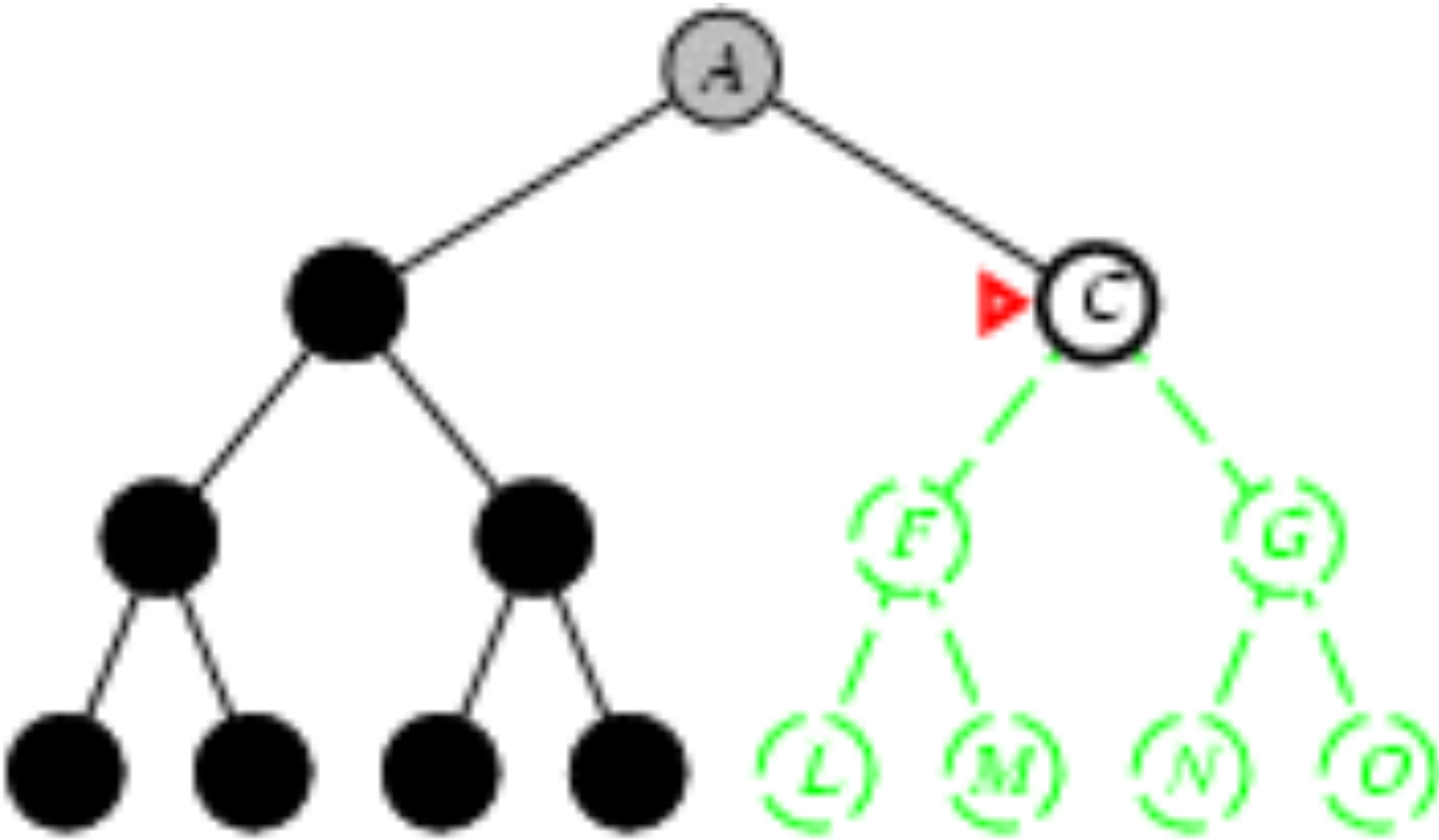

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O

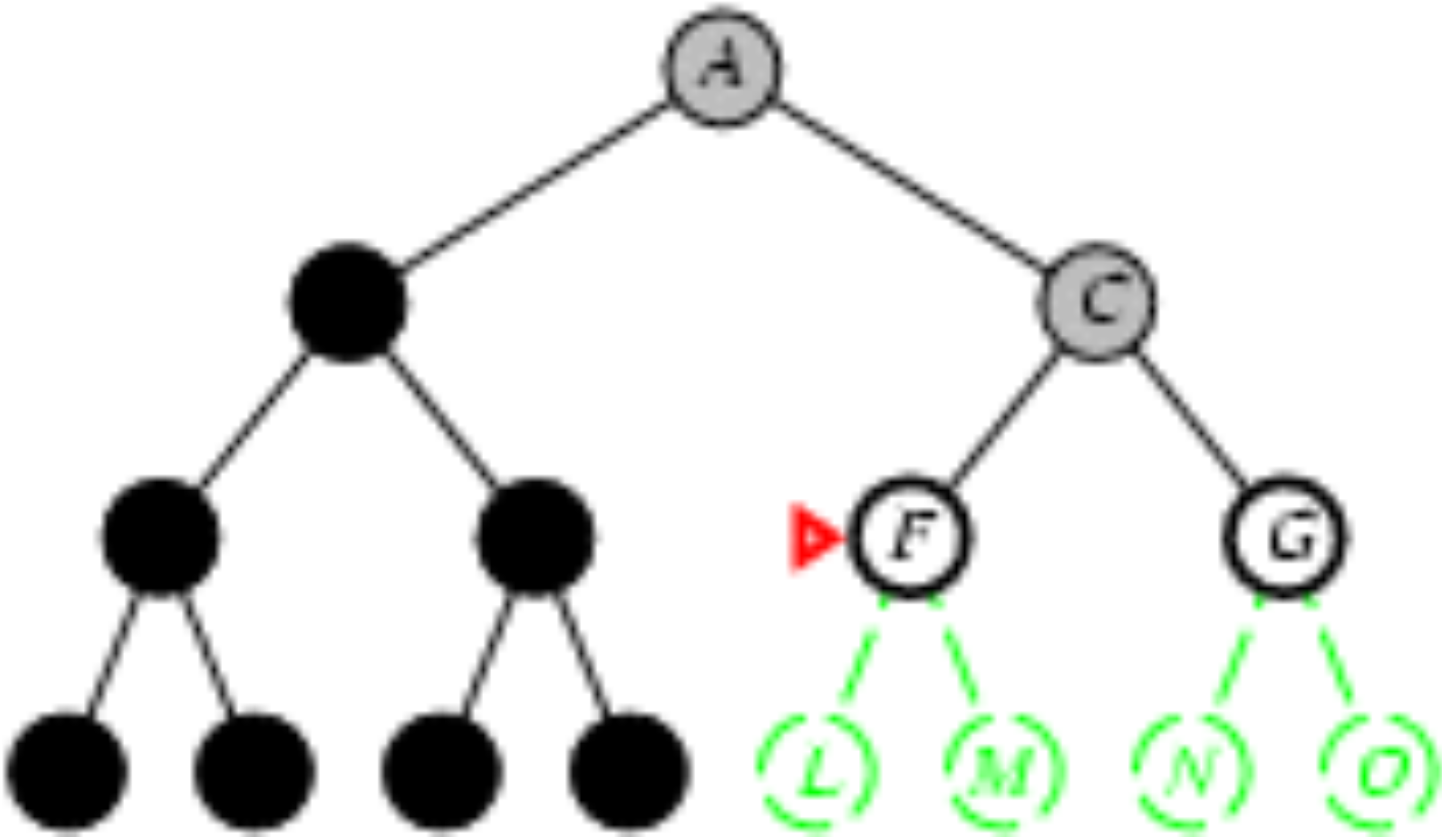

A
B
C
D
E
F
G
I
J
K
L
M
N
O


A
B
C
E
F
G
J
K
L
M
N
O


A
B
C
E
F
G
J
K
L
M
N
O


A
B
C
E
F
G
K
L
M
N
O

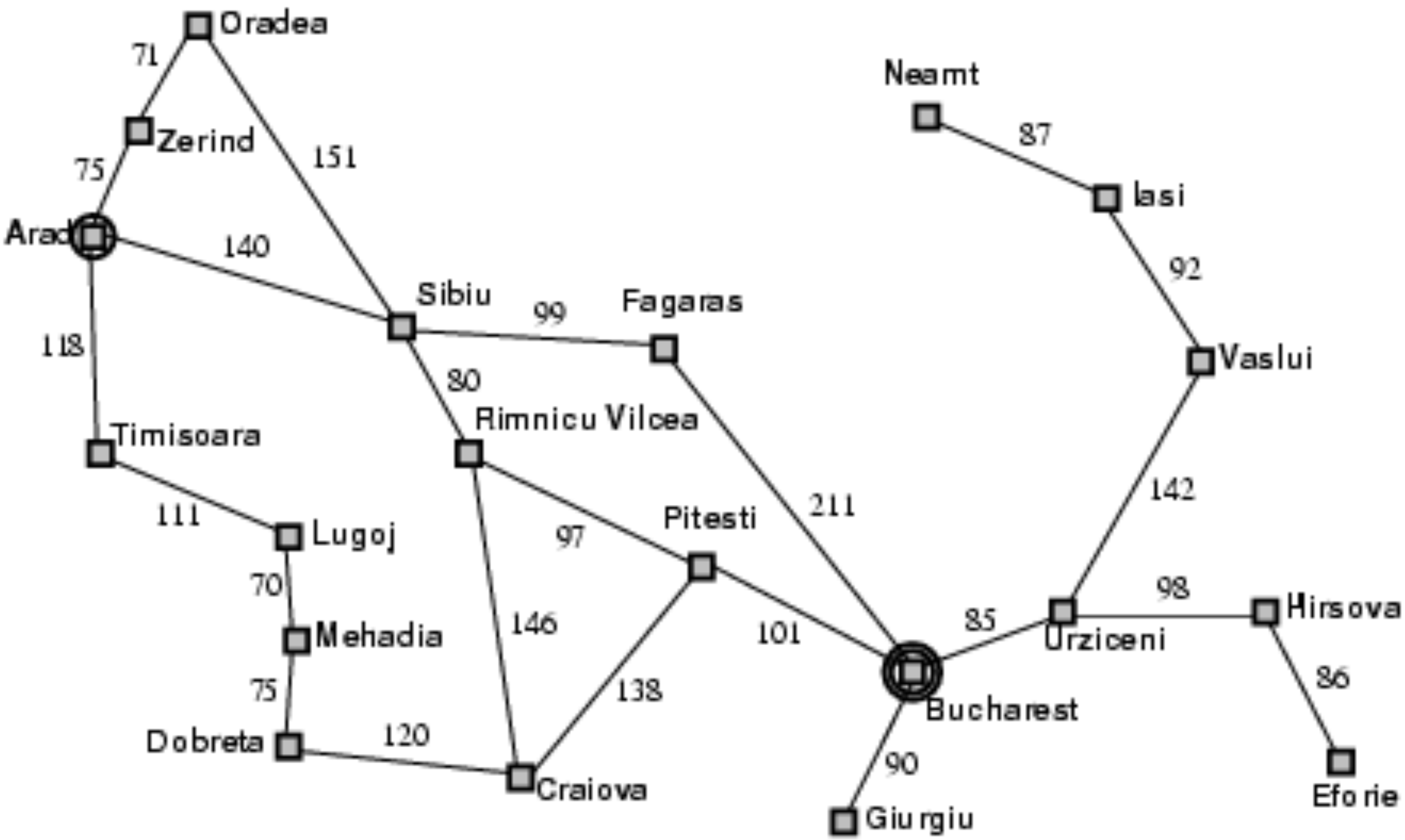

A
C
F
G
L
M
N
O


A
C
F
G
L
M
N
O

Depth-first search

- Complete? No: fails in infinite-depth spaces, spaces with loops
  - Modify to avoid repeated states along path
  - complete in finite spaces
- Time? $O(b^m)$: terrible if m is much larger than d
  - but if solutions are dense, may be much faster than breadth-first
- Space? O(bm): linear space!
- Optimal? No

From Arad to Bucharest



Depth-limited search

- Depth-first search with depth limit l; nodes at depth l have no successors

```
function DEPTH-LIMITED-SEARCH( problem, limit) returns soln/fail/cutoff
    RECURSIVE-DLS(MAKE-NODE(INITIAL-STATE[problem]), problem, limit)

function RECURSIVE-DLS(node, problem, limit) returns soln/fail/cutoff
    cutoff-occurred? ← false
    if GOAL-TEST[problem](STATE[node]) then return SOLUTION(node)
    else if DEPTH[node] = limit then return cutoff
    else for each successor in EXPAND(node, problem) do
        result ← RECURSIVE-DLS(successor, problem, limit)
        if result = cutoff then cutoff-occurred? ← true
        else if result ≠ failure then return result
    if cutoff-occurred? then return cutoff else return failure
```

Iterative deepening

- Do depth-limited search at increasing depths

function ITERATIVE-DEEPENING-SEARCH(*problem*) **returns** a solution, or failure
    inputs: *problem*, a problem
    for *depth* ← 0 **to** ∞ **do**
        result ← DEPTH-LIMITED-SEARCH(*problem*, *depth*)
        if *result* ≠ cutoff **then return** *result*

Summary of algorithms

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening
Complete?	Yes	Yes	No	No	Yes
Time	$O(b^{d+1})$	$O(b^{\lceil C^*/\epsilon \rceil})$	$O(b^m)$	$O(b^l)$	$O(b^d)$
Space	$O(b^{d+1})$	$O(b^{\lceil C^*/\epsilon \rceil})$	$O(bm)$	$O(bl)$	$O(bd)$
Optimal?	Yes	Yes	No	No	Yes