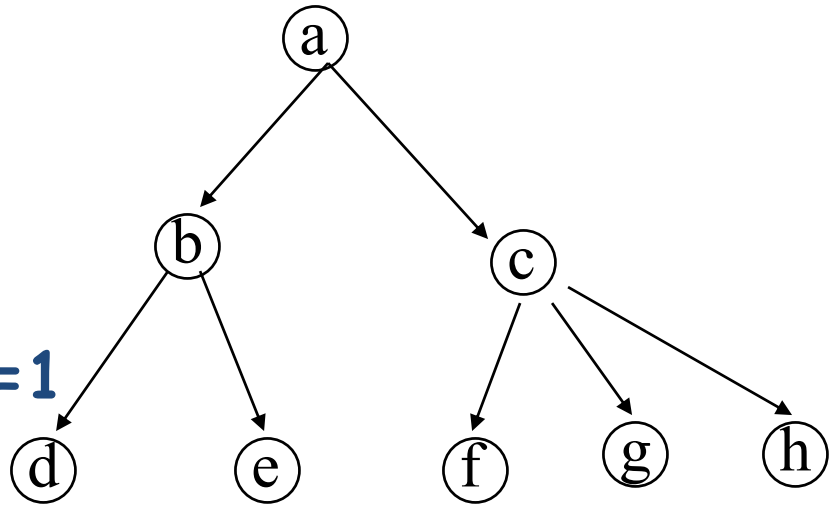
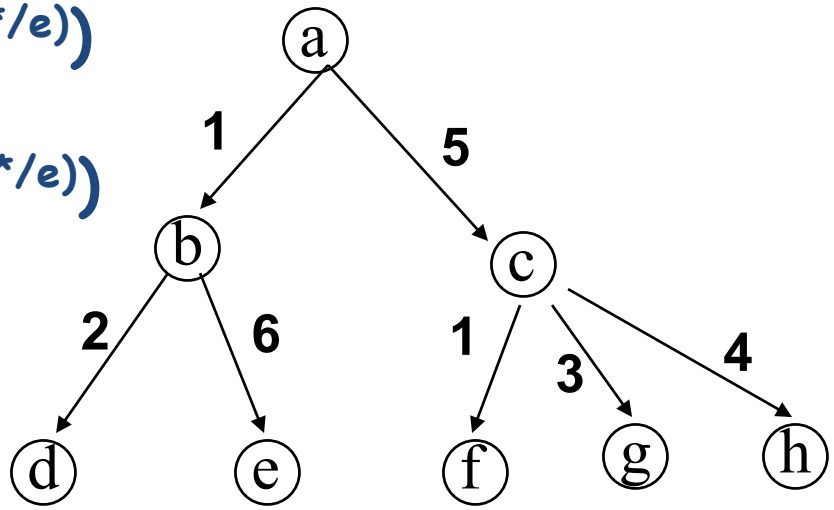
Breadth First Search: shortest first

- Maintain queue of nodes to visit
- Evaluation
  - Complete? **Yes (b is finite)**
  - Time Complexity? **$O(b^d)$**
  - Space Complexity? **$O(b^d)$**
  - Optimal? **Yes, if stepcost=1**



Uniform Cost Search: cheapest first

- Maintain queue of nodes to visit
- Evaluation
  - Complete? Yes (b is finite)
  - Time Complexity? $O(b^{(C^*/e)})$
  - Space Complexity? $O(b^{(C^*/e)})$
  - Optimal? Yes



DFS

http://www.youtube.com/watch?v=dtoFAvtVE4U

UCS

http://www.youtube.com/watch?v=z6lUnb9ktkE

Memory Limitation

- Suppose:
  - 2 GHz CPU
  - 1 GB main memory
  - 100 instructions / expansion
  - 5 bytes / node

  - 200,000 expansions / sec
  - Memory filled in 100 sec … < 2 minutes

Time vs. Memory

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.

Idea 1: Beam Search

- Maintain a constant sized frontier
- Whenever the frontier becomes large
  - Prune the worst nodes

Optimal: no

Complete: no

Idea 2: Iterative deepening search

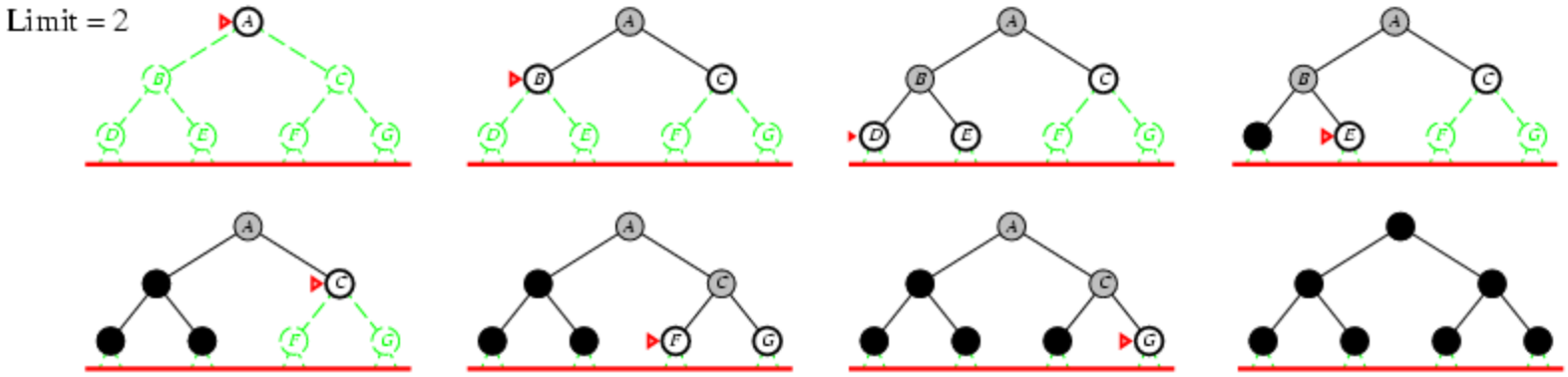
```
function ITERATIVE-DEEPENING-SEARCH( problem) returns a solution, or failure
    inputs: problem, a problem
    for depth ← 0 to ∞ do
        result ← DEPTH-LIMITED-SEARCH( problem, depth)
        if result ≠ cutoff then return result
```

Iterative deepening search *l* =0

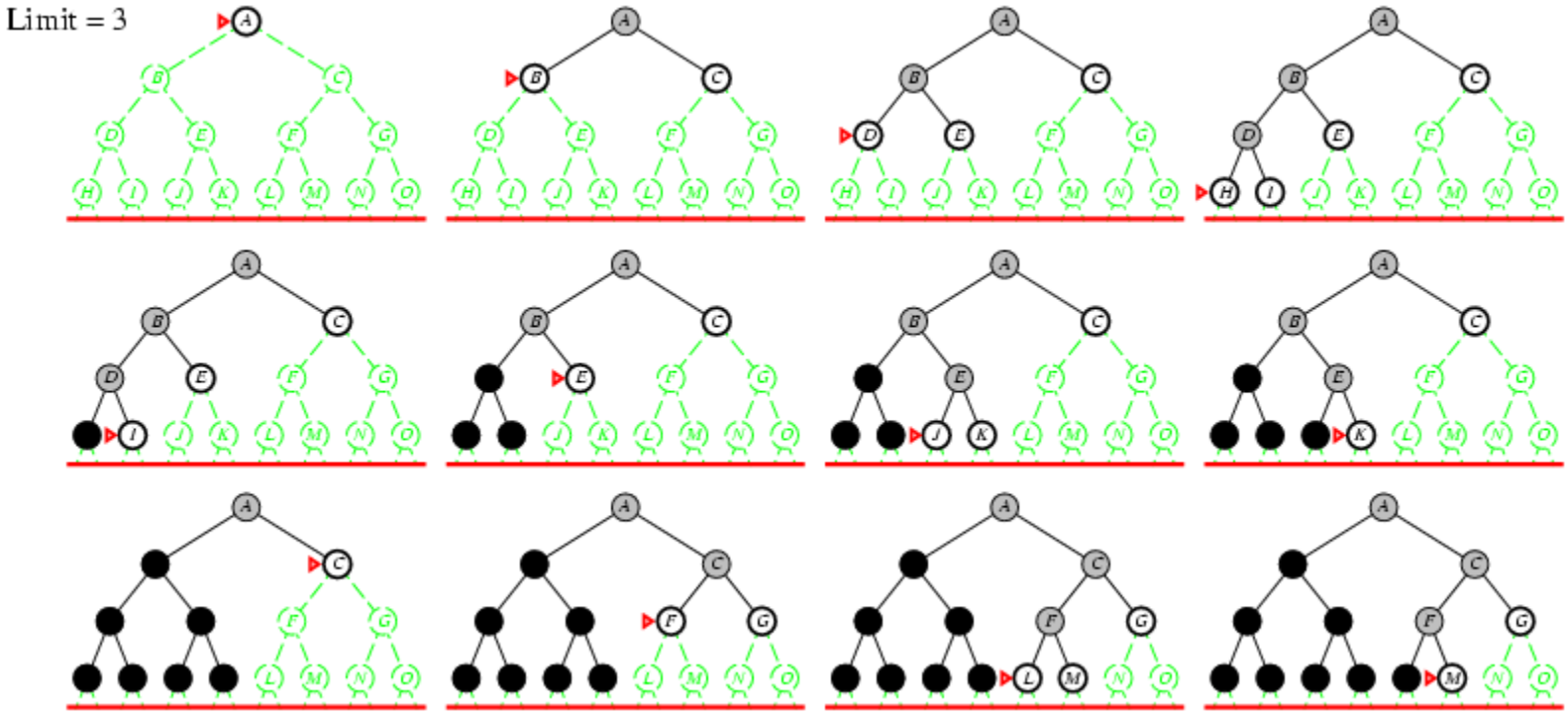
Limit = 0

Iterative deepening search *l* =1

Iterative deepening search *l* =2



Iterative deepening search *l* =3



Iterative deepening search

- Number of nodes generated in a depth-limited search to depth *d* with branching factor *b*:
  - $N_{DLS} = b^0 + b^1 + b^2 + \ldots + b^{d-2} + b^{d-1} + b^d$
- Number of nodes generated in an iterative deepening search to depth *d* with branching factor *b*:
  - $N_{IDS} = (d+1)b^0 + d\ b\text{^}1 + (d-1)b\text{^}2 + \ldots + 3b^{d-2} + 2b^{d-1} + 1b^d$
- Asymptotic ratio: (b+1)/(b-1)
- For *b = 10*, *d = 5*,
  - $N_{DLS} = 1 + 10 + 100 + 1{,}000 + 10{,}000 + 100{,}000 = 111{,}111$
  - $N_{IDS} = 6 + 50 + 400 + 3{,}000 + 20{,}000 + 100{,}000 = 123{,}456$
- Overhead = (123,456 - 111,111)/111,111 = 11%

Iterative deepening search

- Complete?
  - Yes
- Time?
  - $(d+1)b^0 + d\,b^1 + (d-1)b^2 + \ldots + b^d = O(b^d)$
- Space?
  - $O(bd)$
- Optimal?
  - Yes, if step cost = 1
  - Can be modified to explore uniform cost tree (iterative lengthening)
- Systematic?