Search (complementary slides)

Tomas Svoboda, BE5B33KUI
2017-03-13

Slide material from CS 188: Artificial Intelligence at UCB
by Dan Klein, and Pieter Abbeel, used with permision

BE5B33KUI - admin

- Deadline for the 02_search postponed to March 20 (night after the next computer lab)
- The lecture program will change slightly - due to the delayed start

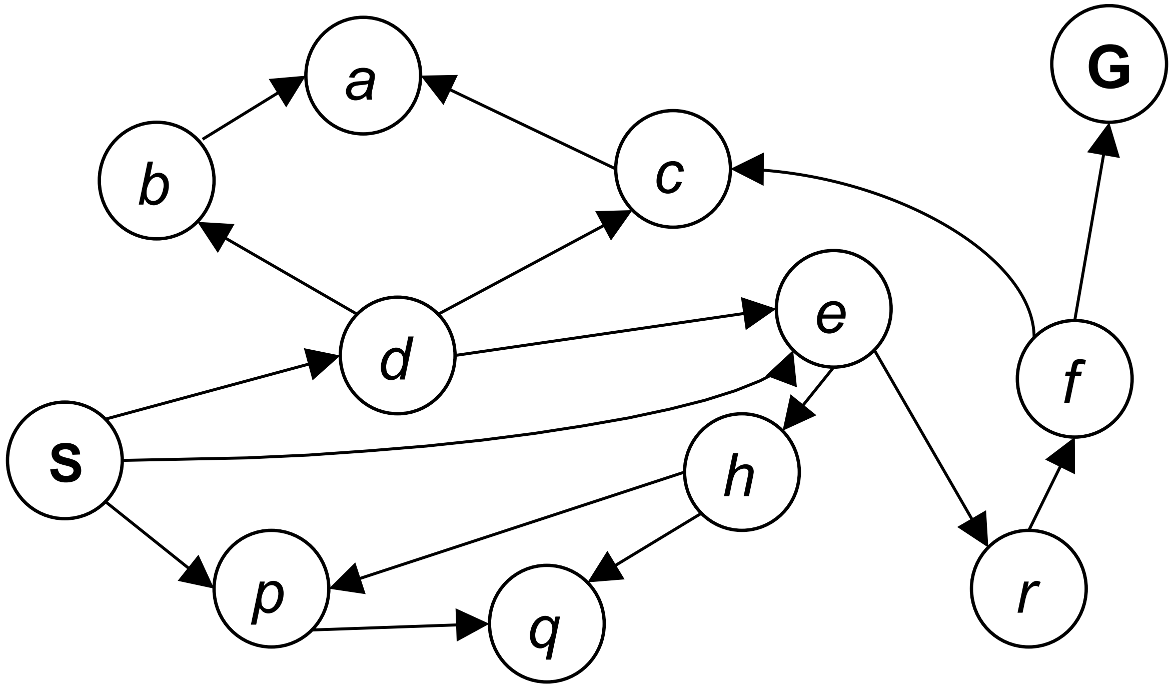
Graphs and Trees

State Space Graphs

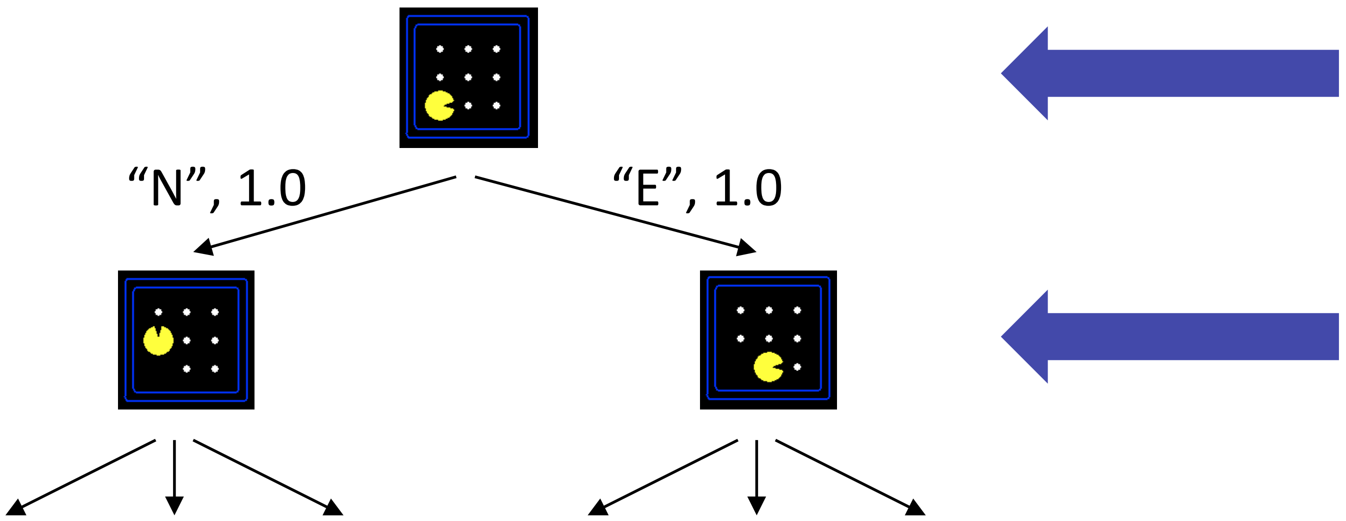
- State space graphs - mathematical representation of a search problem

- Every state only once

- Do we need the whole graph? (in a memory)



Search Trees


“N”, 1.0

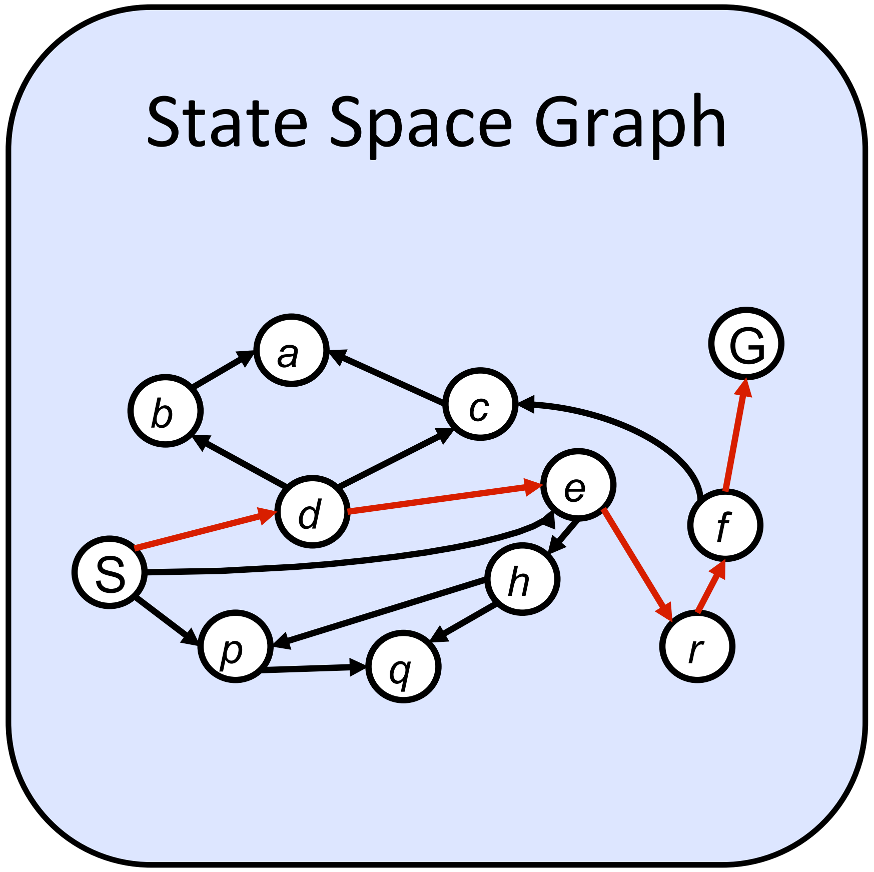
“E”, 1.0

This is now / start

Possible futures

- A “what-if” tree of plans and outcomes
- Parents, children
- Nodes of the tree contain states, but much more

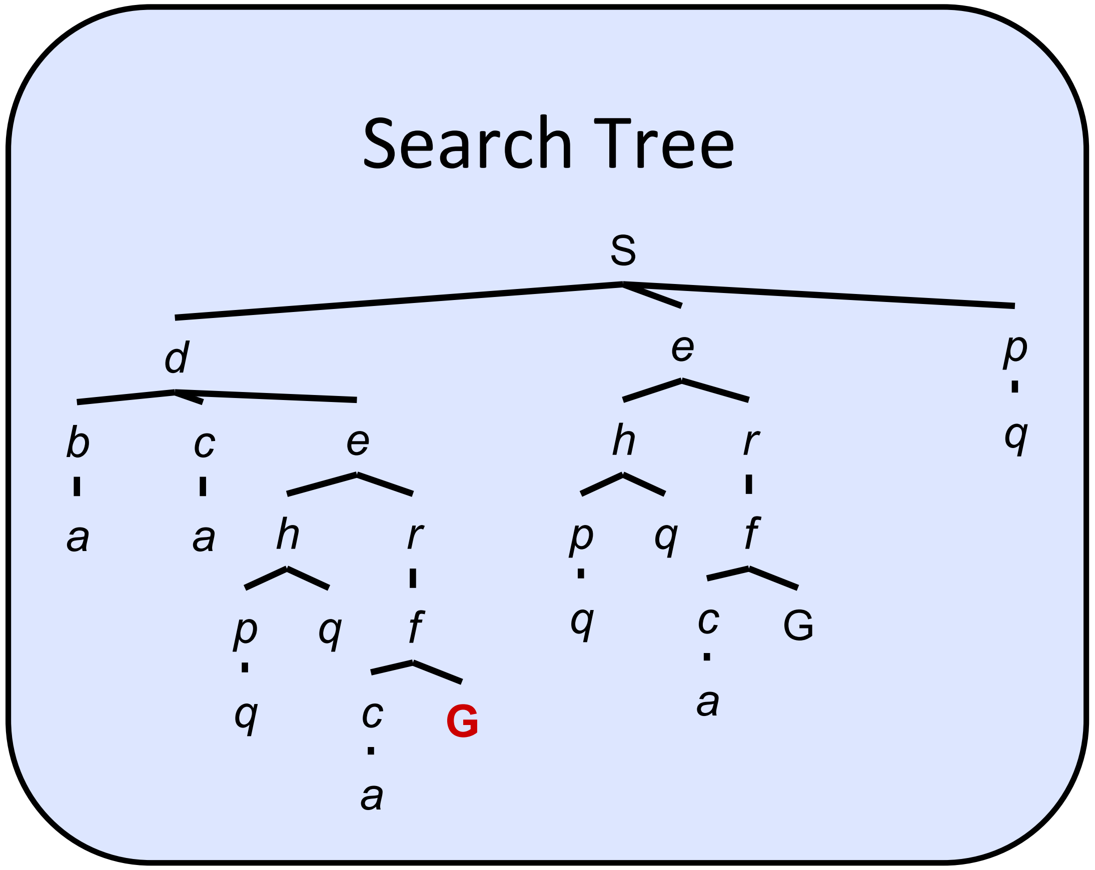
State Space Graphs vs. Search Trees



State Space Graph

Each NODE in in the search tree is an entire PATH in the problem graph.

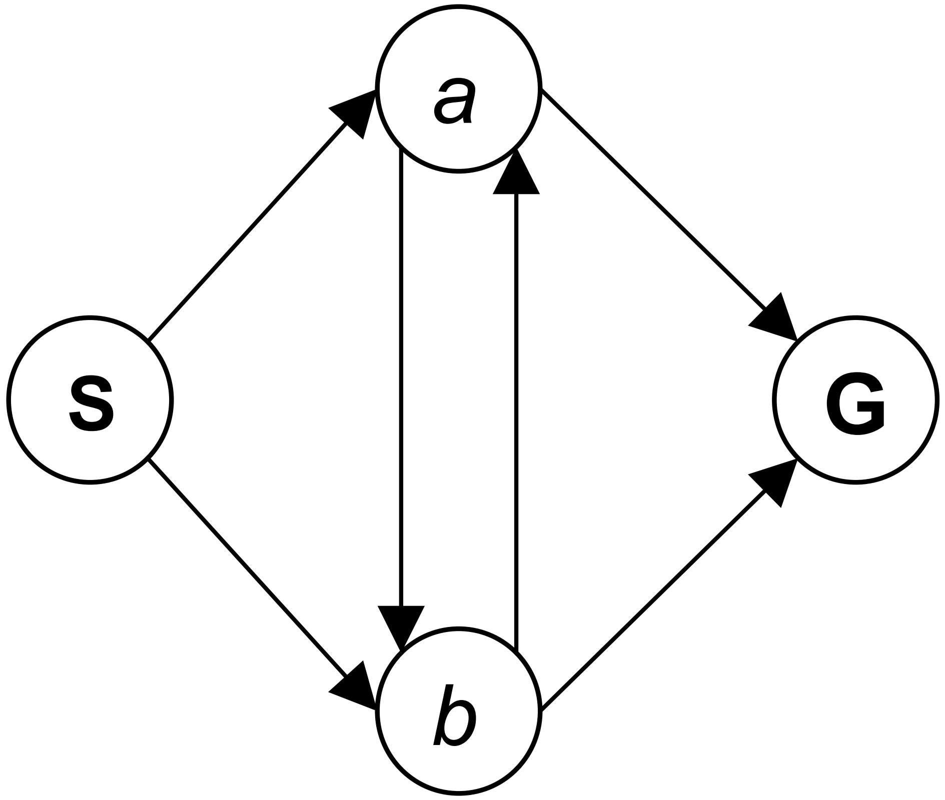
We construct both on demand – and we construct as little as possible.



Search Tree

State Graphs vs. Search Trees

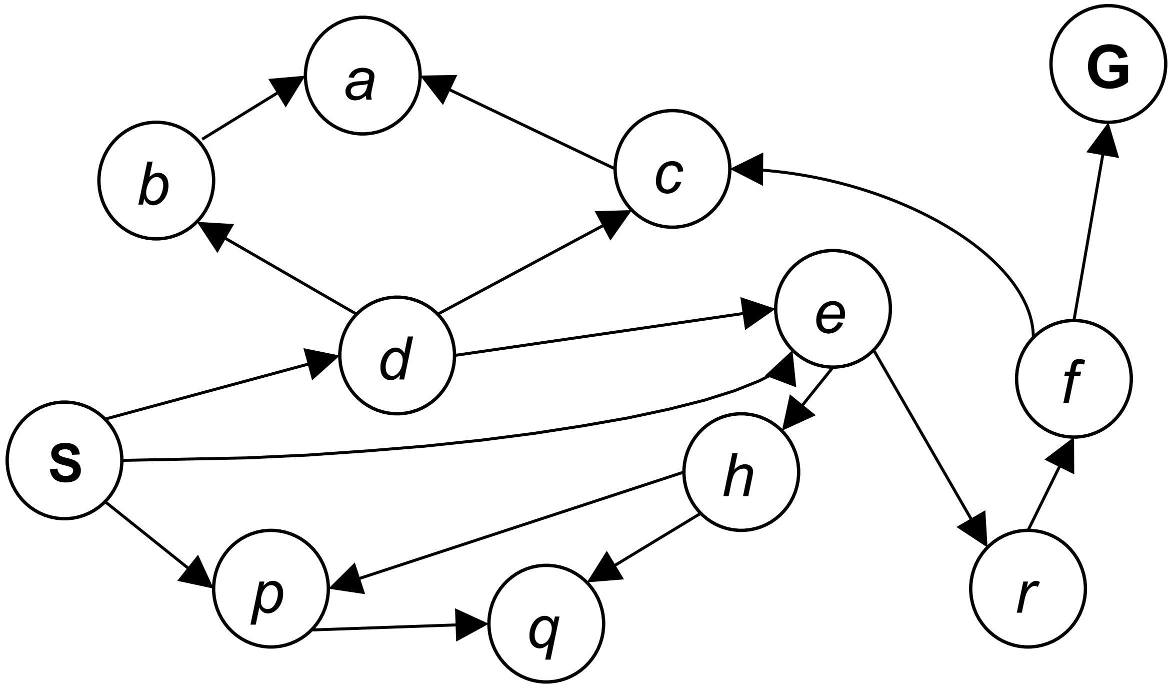
A 4-state graph



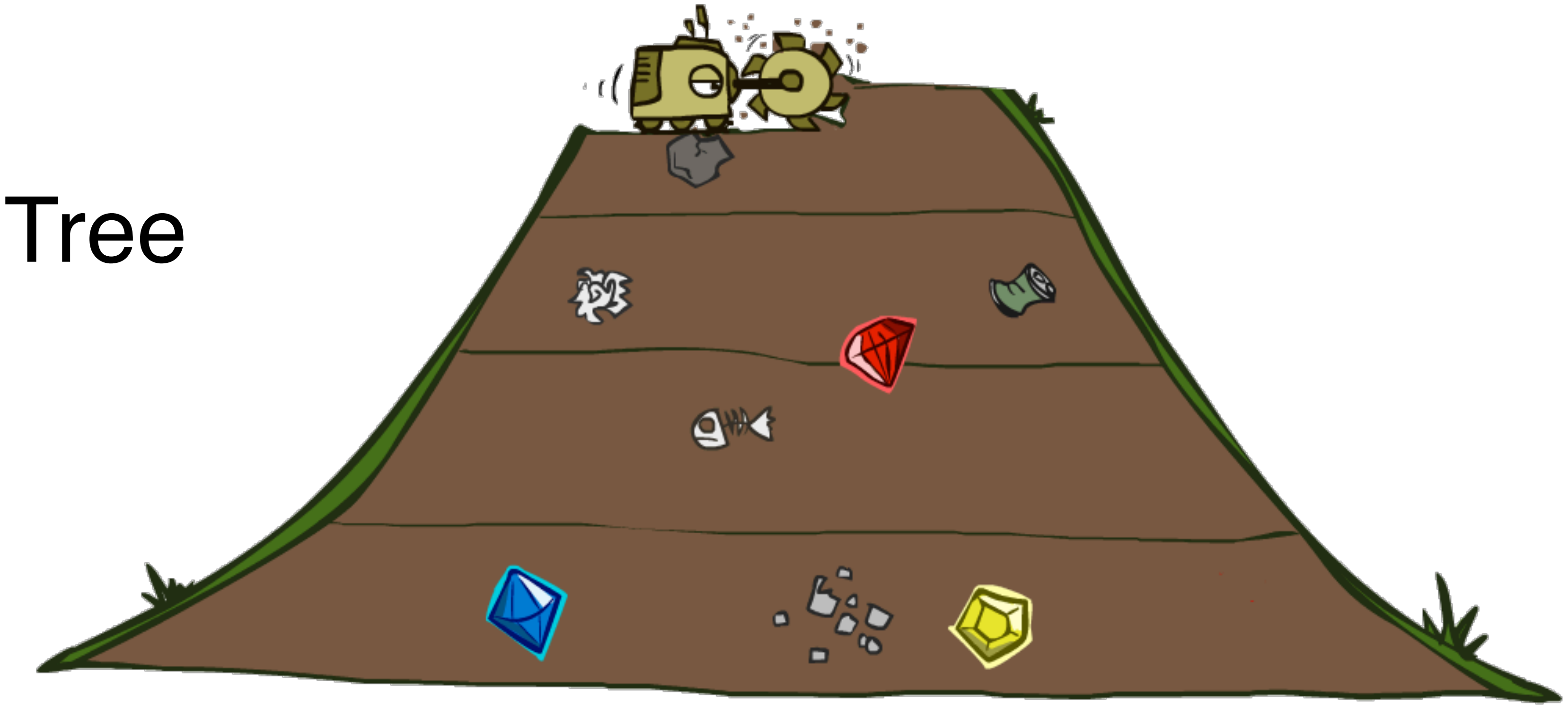
How big is the search tree?

∞

Breadth-First Search (BFS)



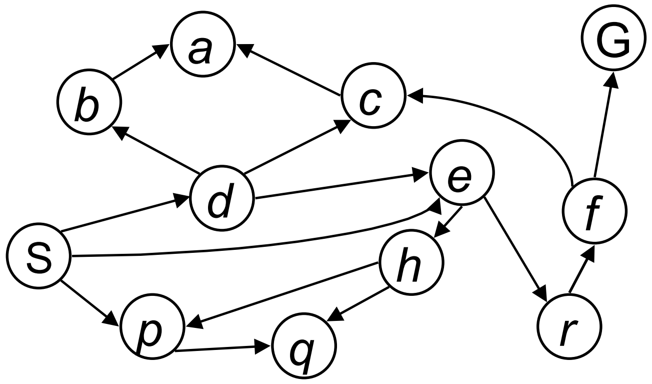
Now, build the Search Tree



Breadth-First Search

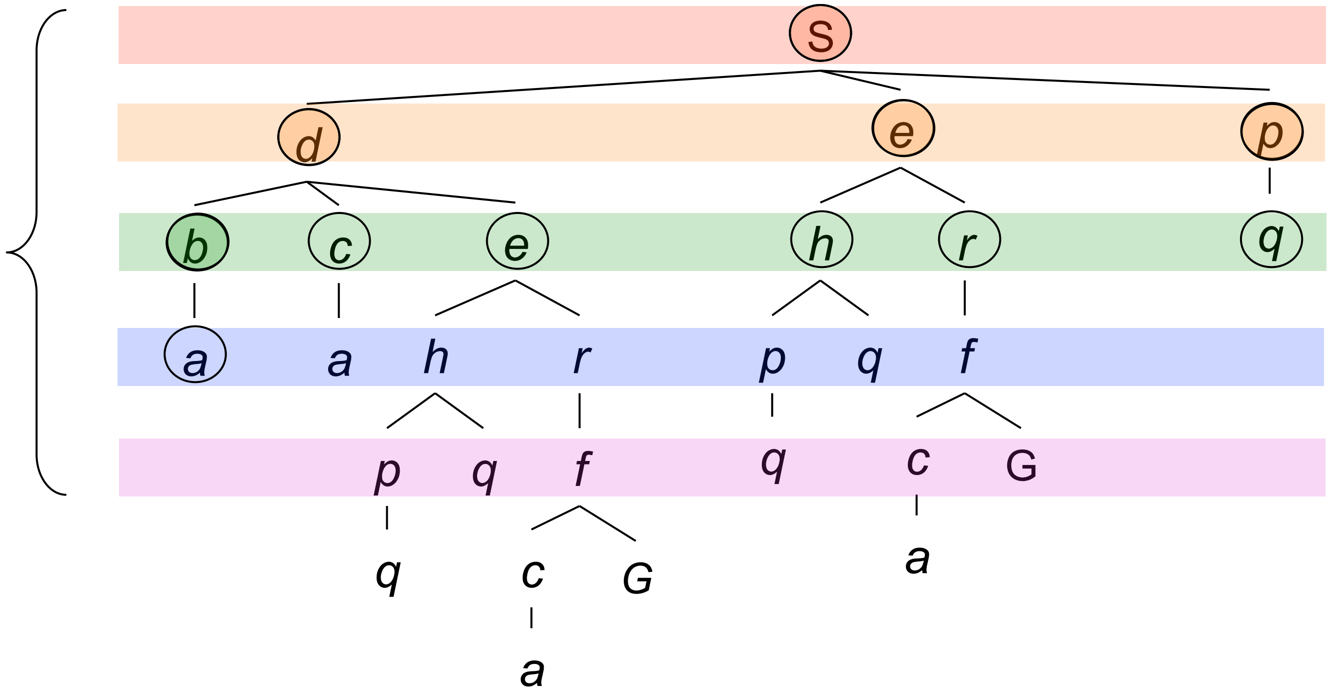
Strategy: expand a shallowest node first

Implementation: Fringe is a FIFO queue



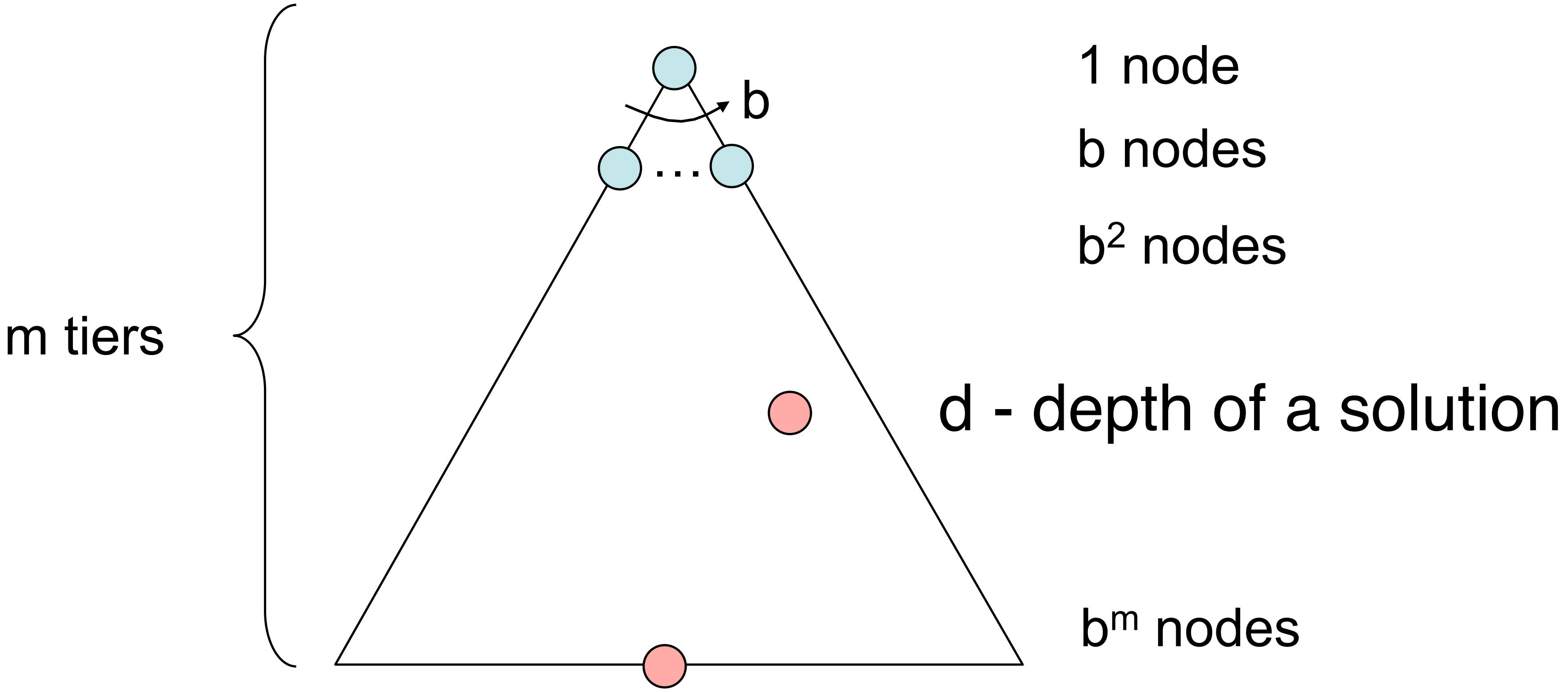
Search

Tiers

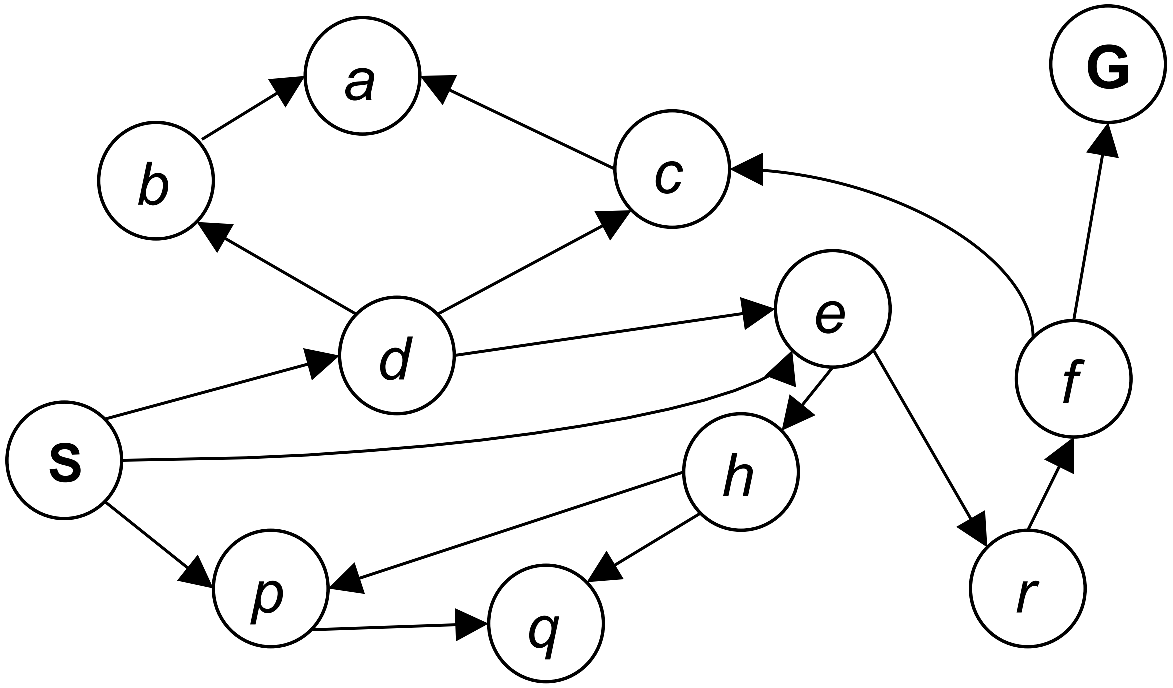


BFS properties

- Time?
- Space?
- Complete?
- Optimal?



Depth-First Search (DFS)

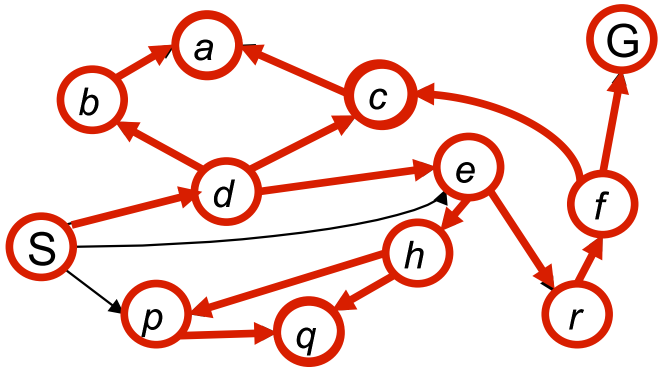


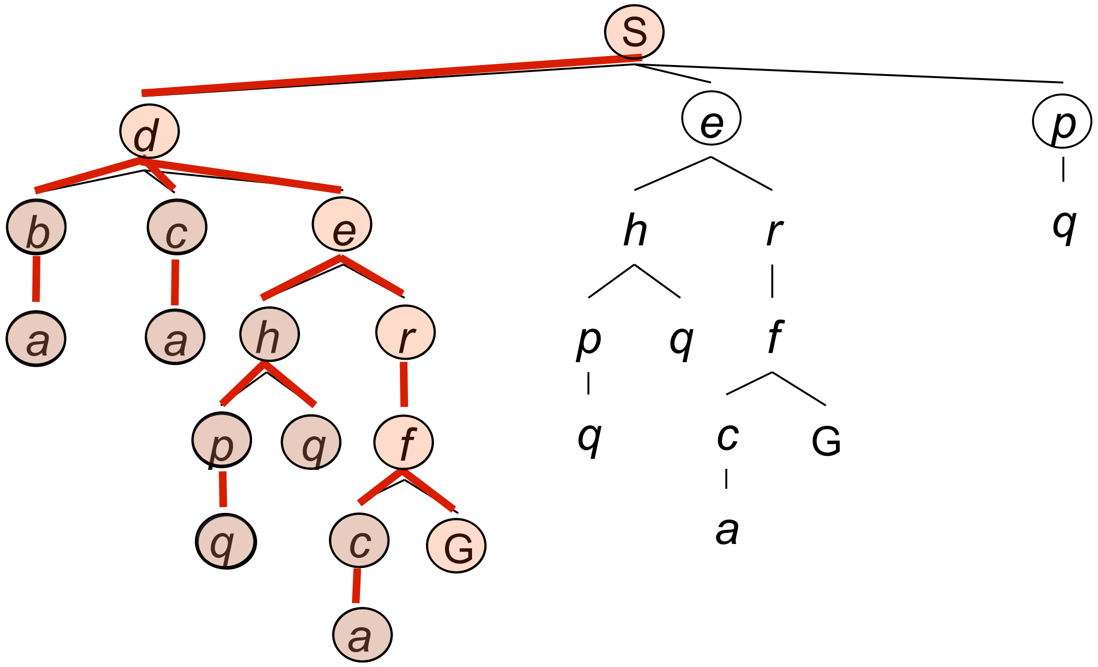
Now, build the Search Tree

Depth-First Search

Strategy: expand a deepest node first

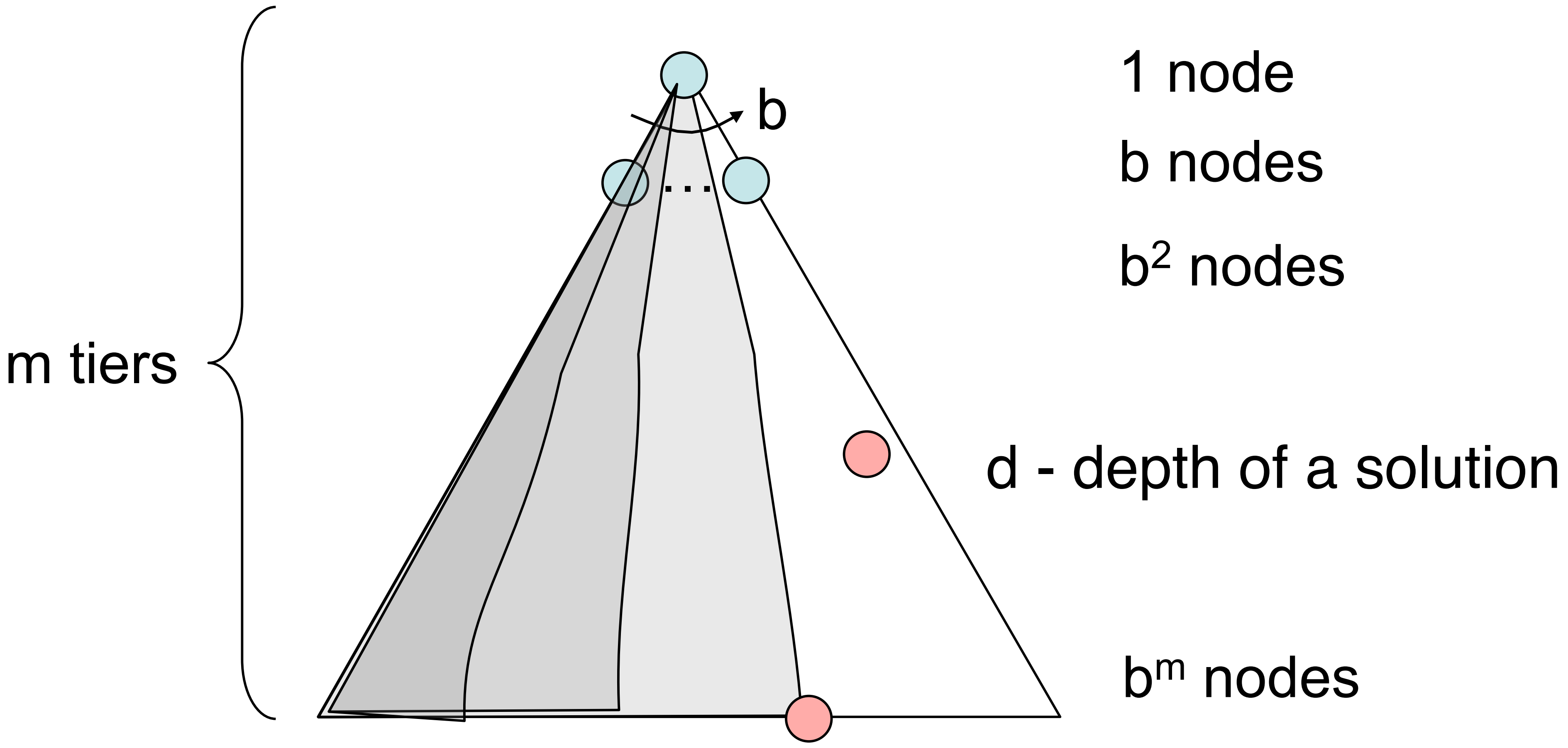
Implementation: Fringe is a LIFO stack



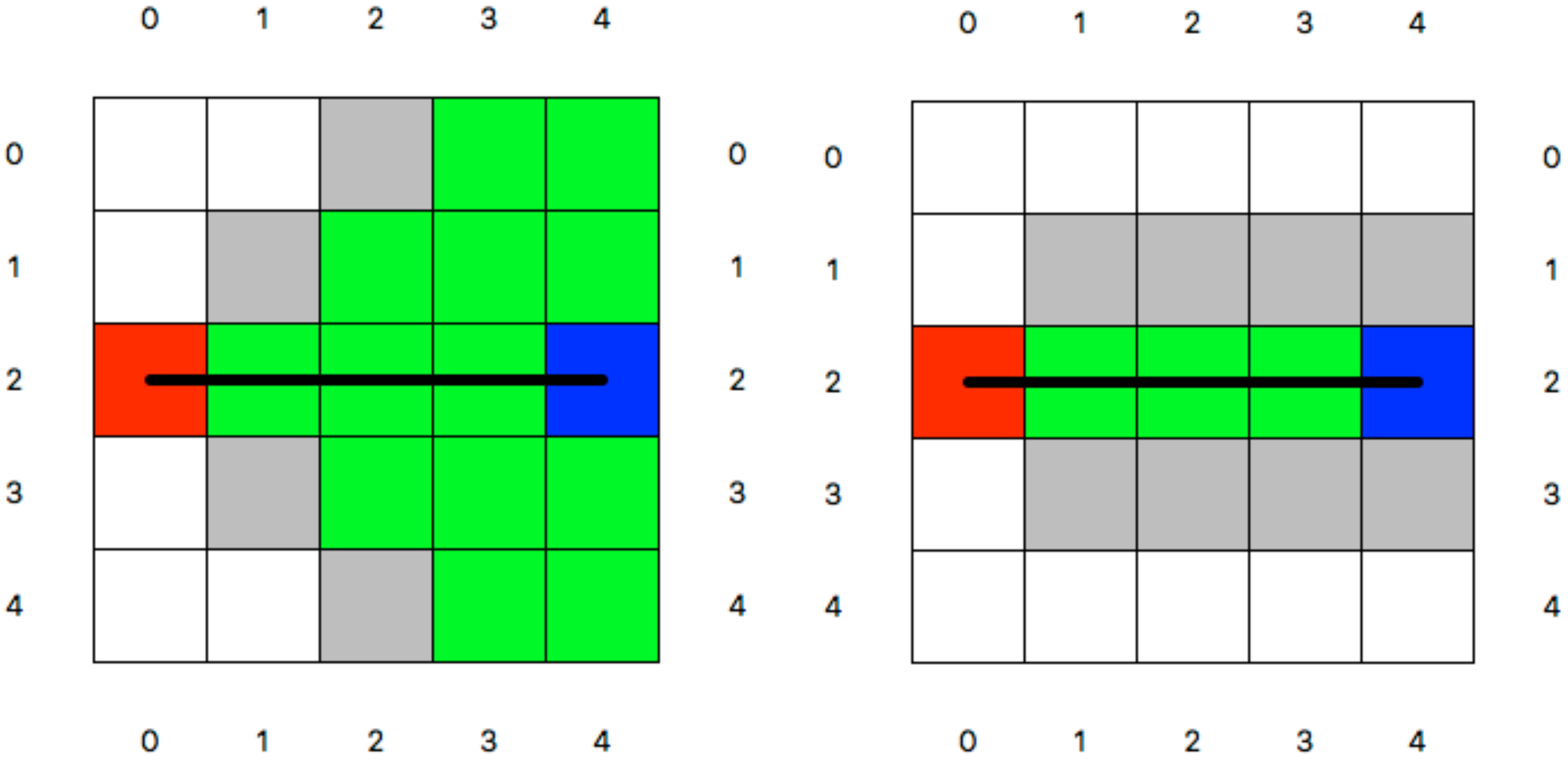


DFS properties

- Time?
- Space?
- Complete?
- Optimal?

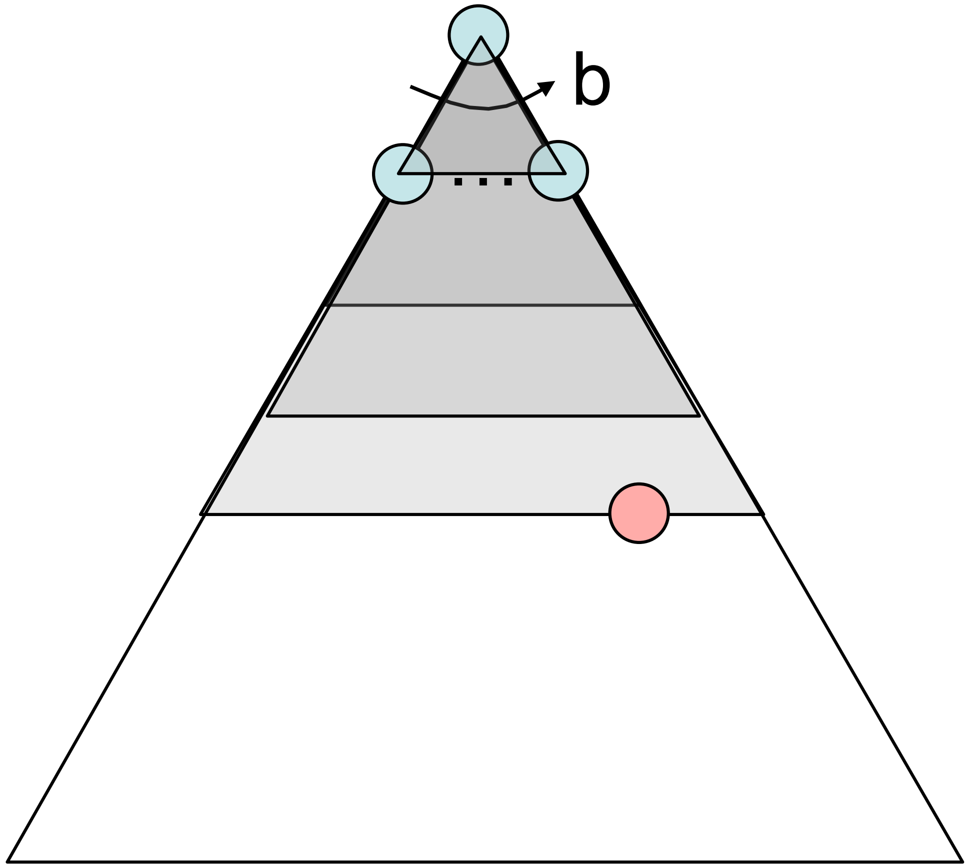


BFS vs DFS



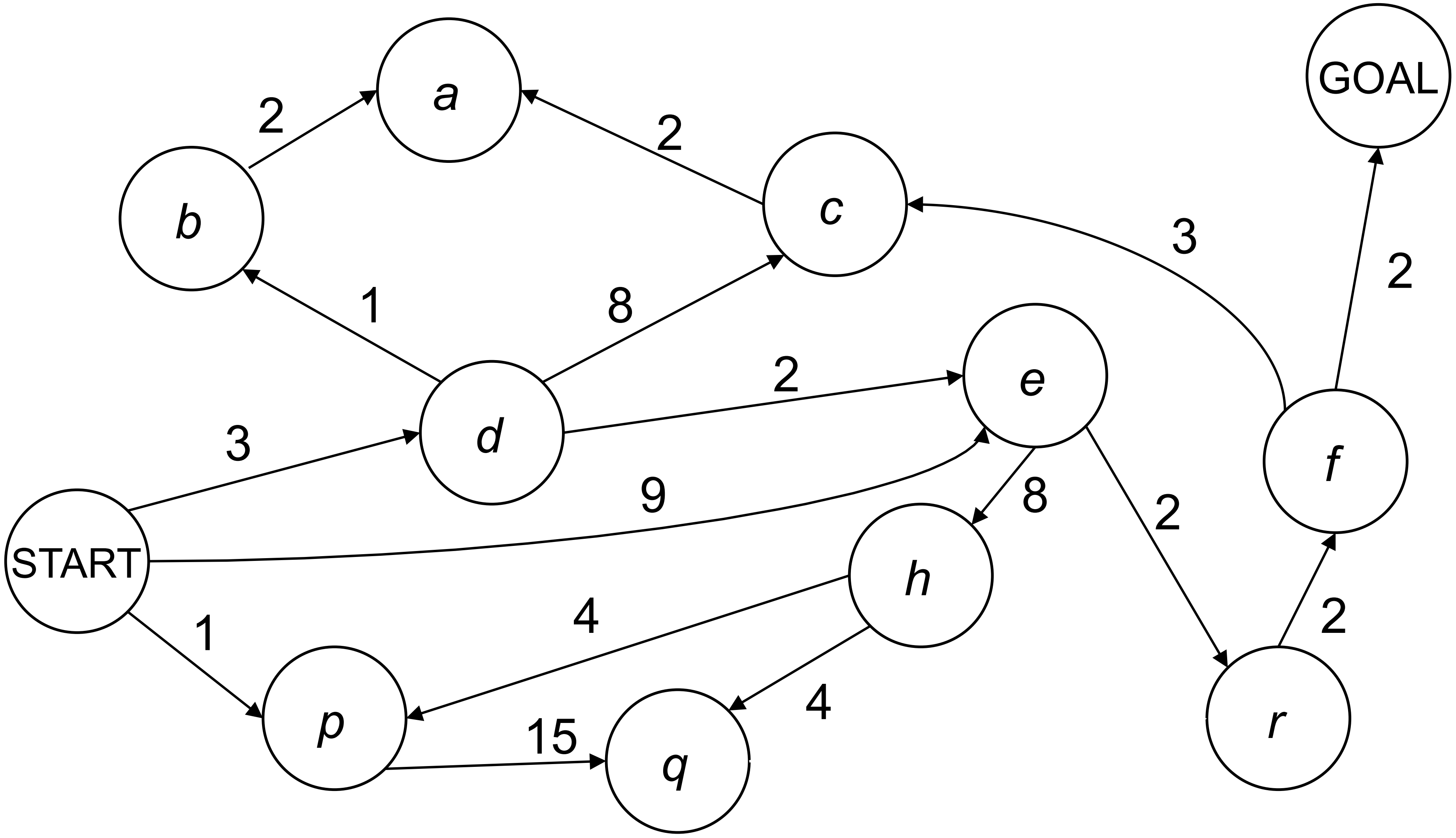
Iterative Deepening Search

- Run a DFS with depth limit 1
- If no solution
- Run a DFS with depth limit 2
- …



Is it not too wasteful? Even compared to the BFS?

Cost-sensitive search

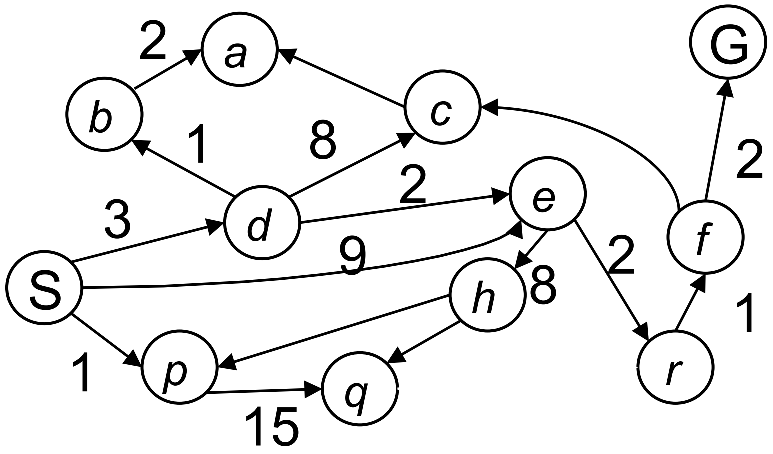


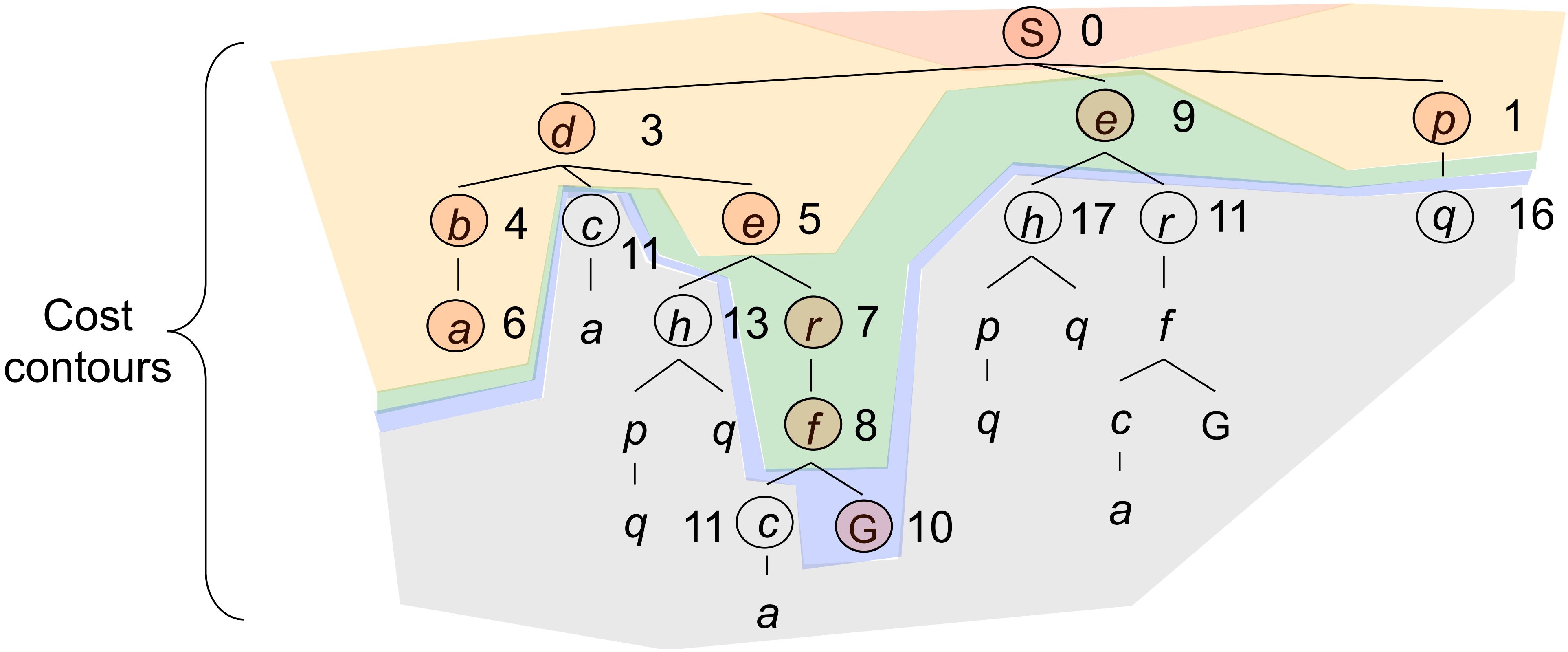
What about using the BFS?

Uniform Cost Search (UCS)

Strategy: expand a cheapest node first:

Fringe is a priority queue (priority: cumulative cost)

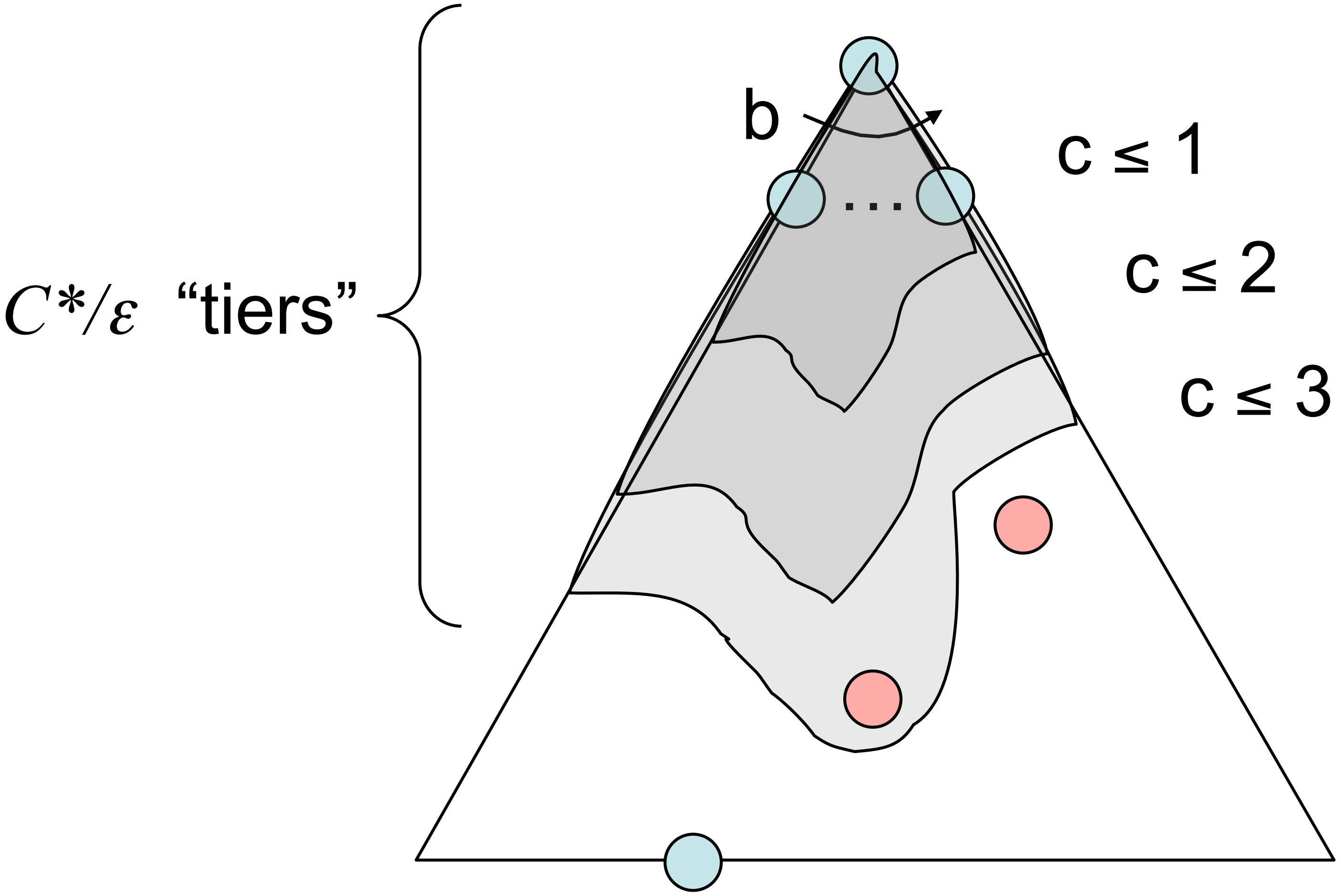




UCS properties

- C^* - solution cost
- ϵ - arc minimum cost

- Time
- Space
- Complete?
- Optimal?



The One Queue

- All algorithms the same …
- except the fringe strategies