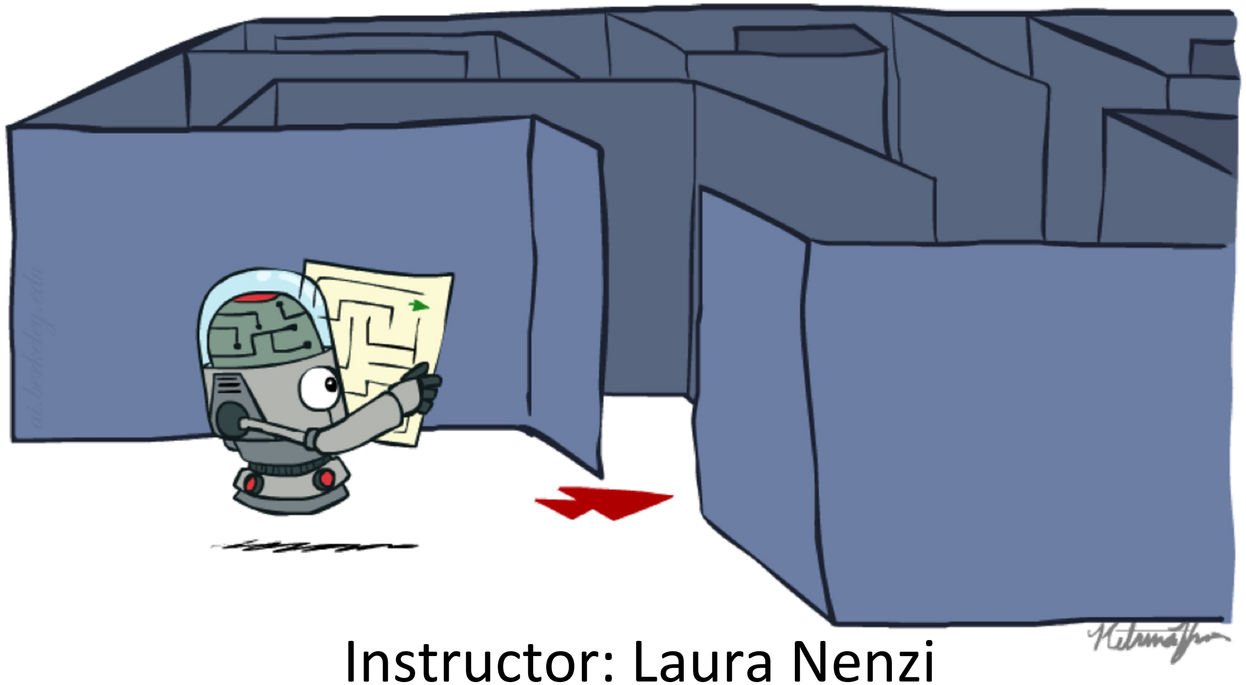
Introduction to Artificial Intelligence

Uninformed Search



Instructor: Laura Nenzi

University of Trieste, Italy

[slides adapted from Dan Klein, Pieter Abbeel, Stuart Russell, et al for CS188 Intro to AI at UC Berkeley. All materials available at http://ai.berkeley.edu.]

Uninformed Search

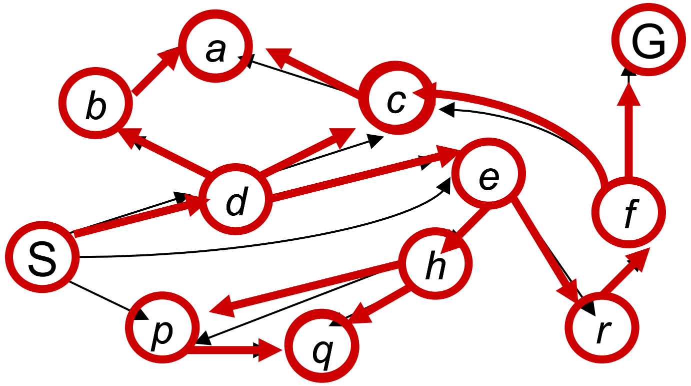
No clue about how close a state is to the goal(s)

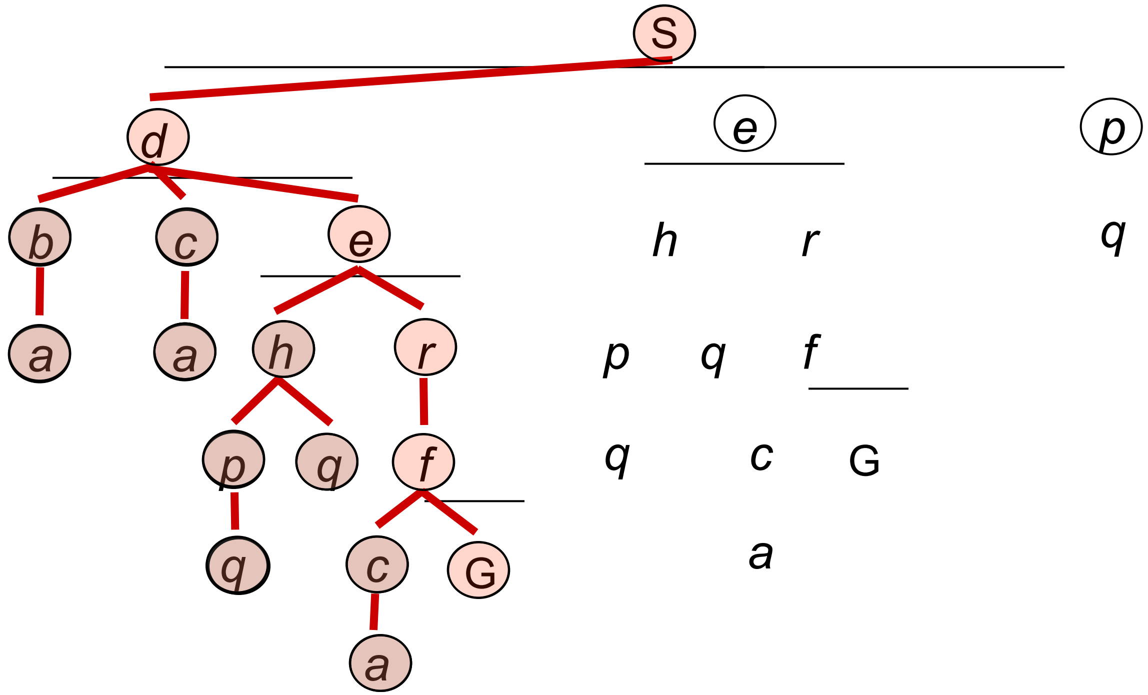
Depth-First Search (DFS)

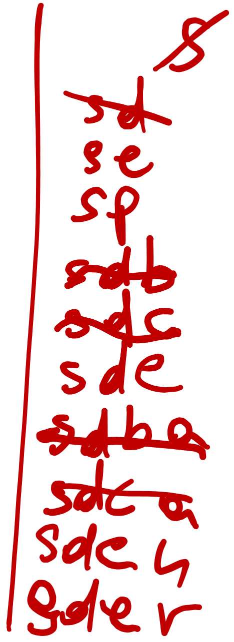
Depth-First Search

Strategy: expand a deepest node first

Implementation: Fringe is a LIFO stack

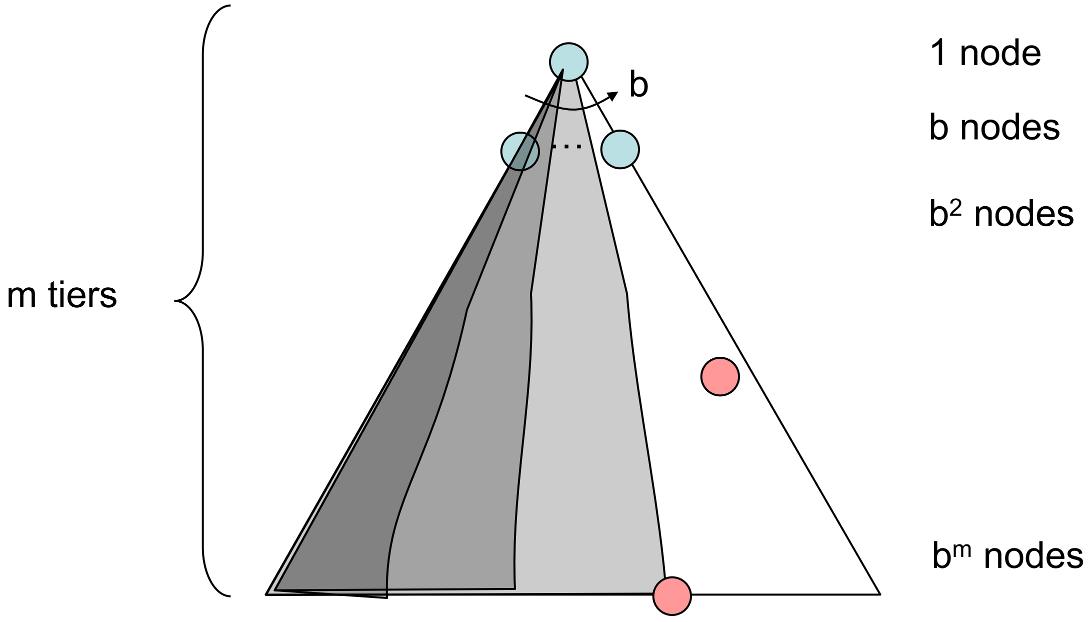




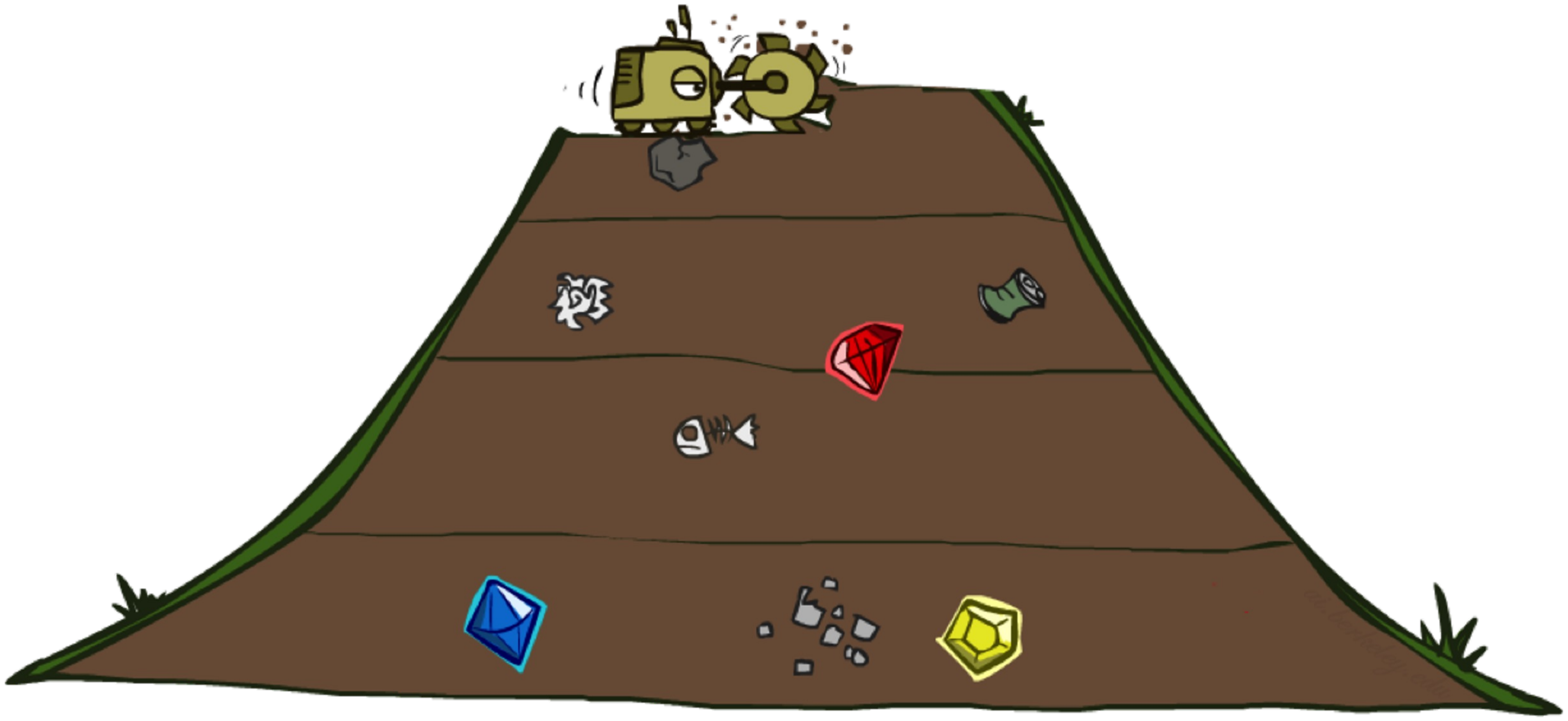


Depth-First Search (DFS) Properties

- What nodes DFS expand?
  - Some left prefix of the tree.
  - Could process the whole tree!
  - If m is finite, takes time $O(b^m)$
- How much space does the fringe take?
  - Only has siblings on path to root, so $O(bm)$
- Is it complete?
  - m could be infinite, so only if we prevent cycles (more later)
- Is it optimal?
  - No, it finds the "leftmost" solution, regardless of depth or cost



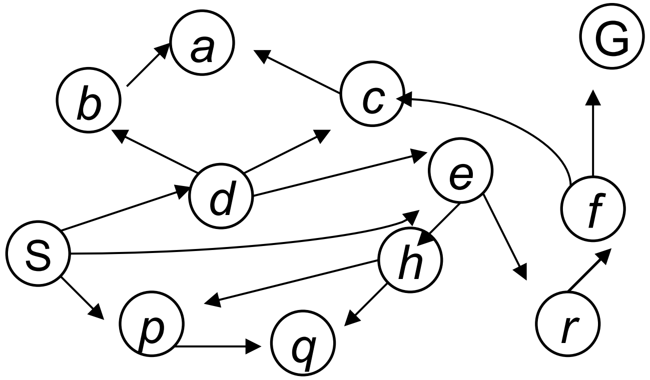
Breadth-First Search (BFS)

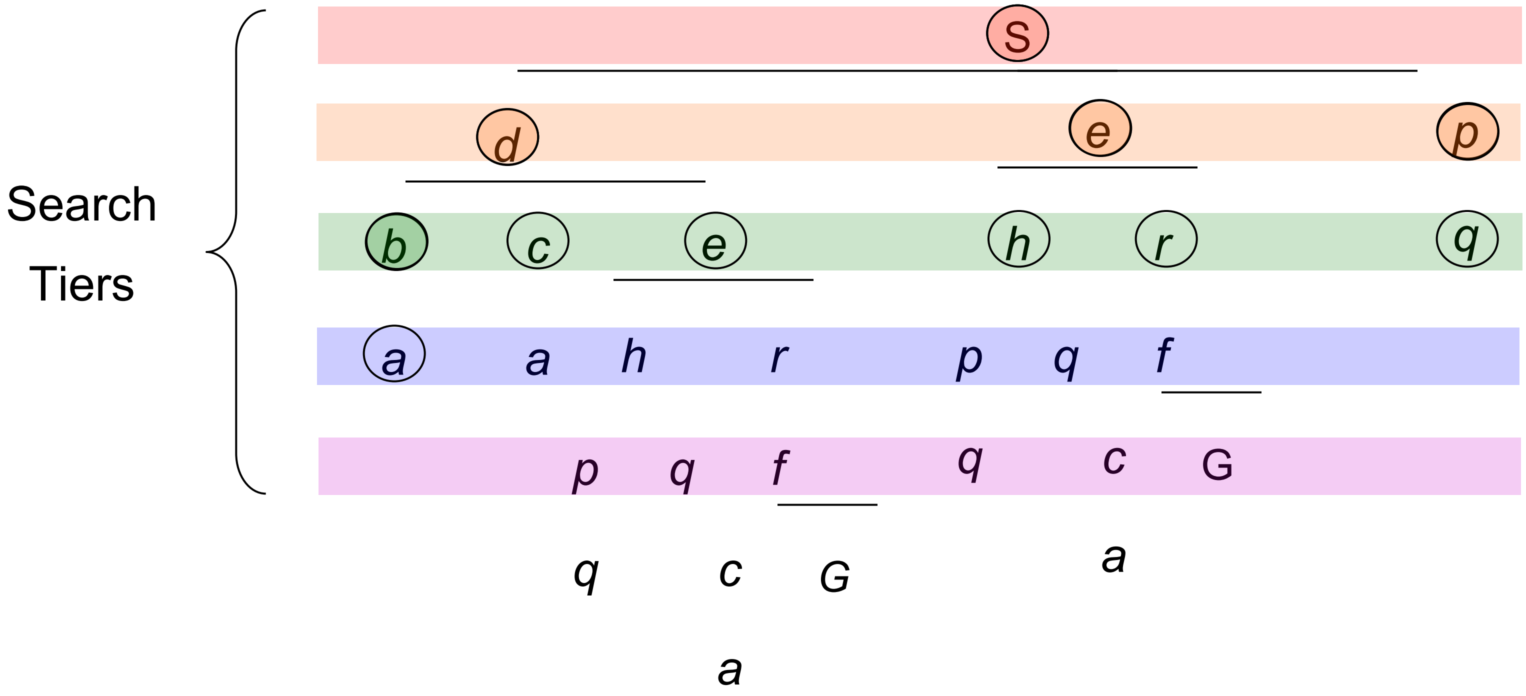


Breadth-First Search

Strategy: expand a shallowest node first

Implementation: Fringe is a FIFO queue

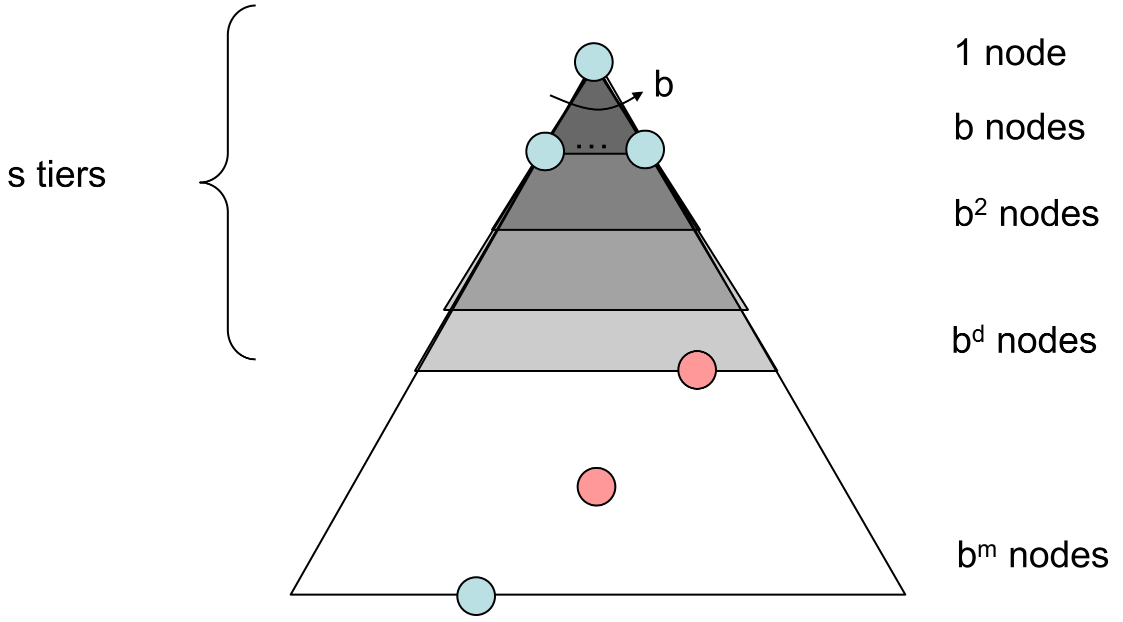




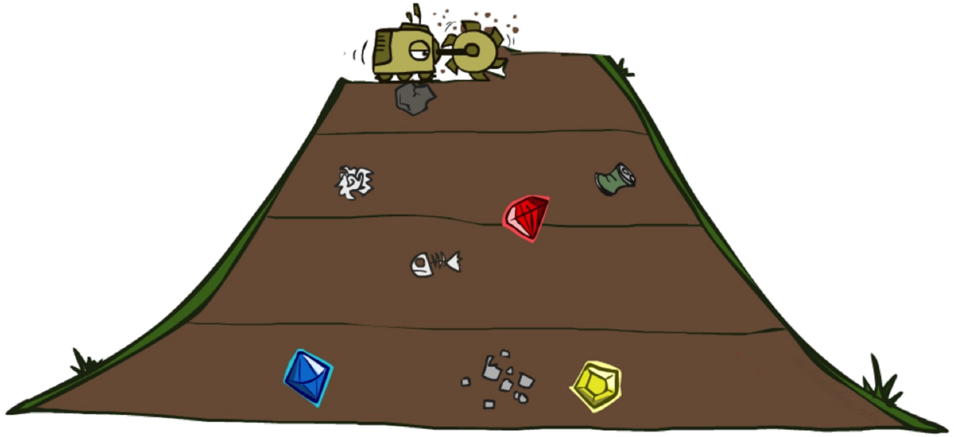
s
sd
se
sp
sdb
sdc
sde
seh
ser
spq

Breadth-First Search (BFS) Properties

- What nodes does BFS expand?
  - Processes all nodes above shallowest solution
  - Let depth of shallowest solution be s
  - Search takes time $O(b^d)$
- How much space does the fringe take?
  - Has roughly the last tier, so $O(b^d)$
- Is it complete?
  - d must be finite if a solution exists, so yes!
- Is it optimal?
  - Only if costs are all 1 (more on costs later)



Quiz: DFS vs BFS



Quiz: DFS vs BFS

- When will BFS outperform DFS?
- When will DFS outperform BFS?

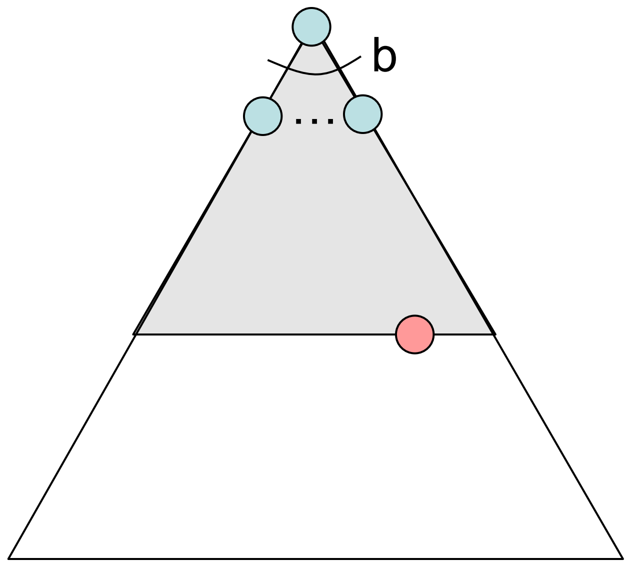
Video of Demo Maze Water DFS/BFS (part 1)

Video of Demo Maze Water DFS/BFS (part 2)

Depth-limited Search (DLS)

- Idea: supply a depth limit ℓ and treat all nodes at depth ℓ as if they had no successors

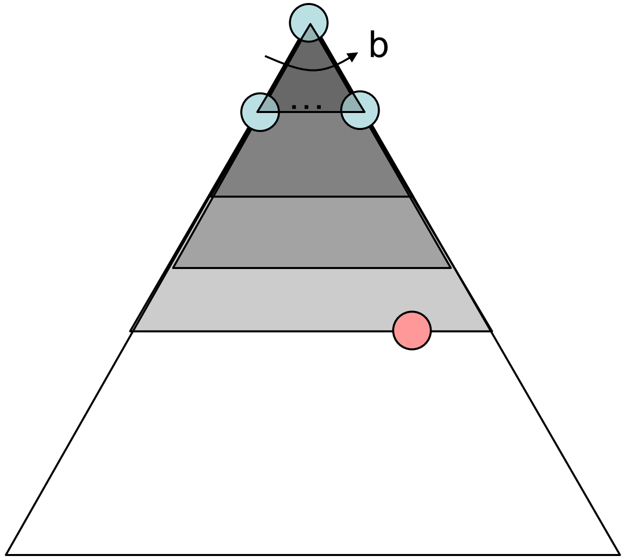
- Time complexity?
  - Search takes time $O(b^{\ell})$
- Space complexity?
  - $O(b\ell)$



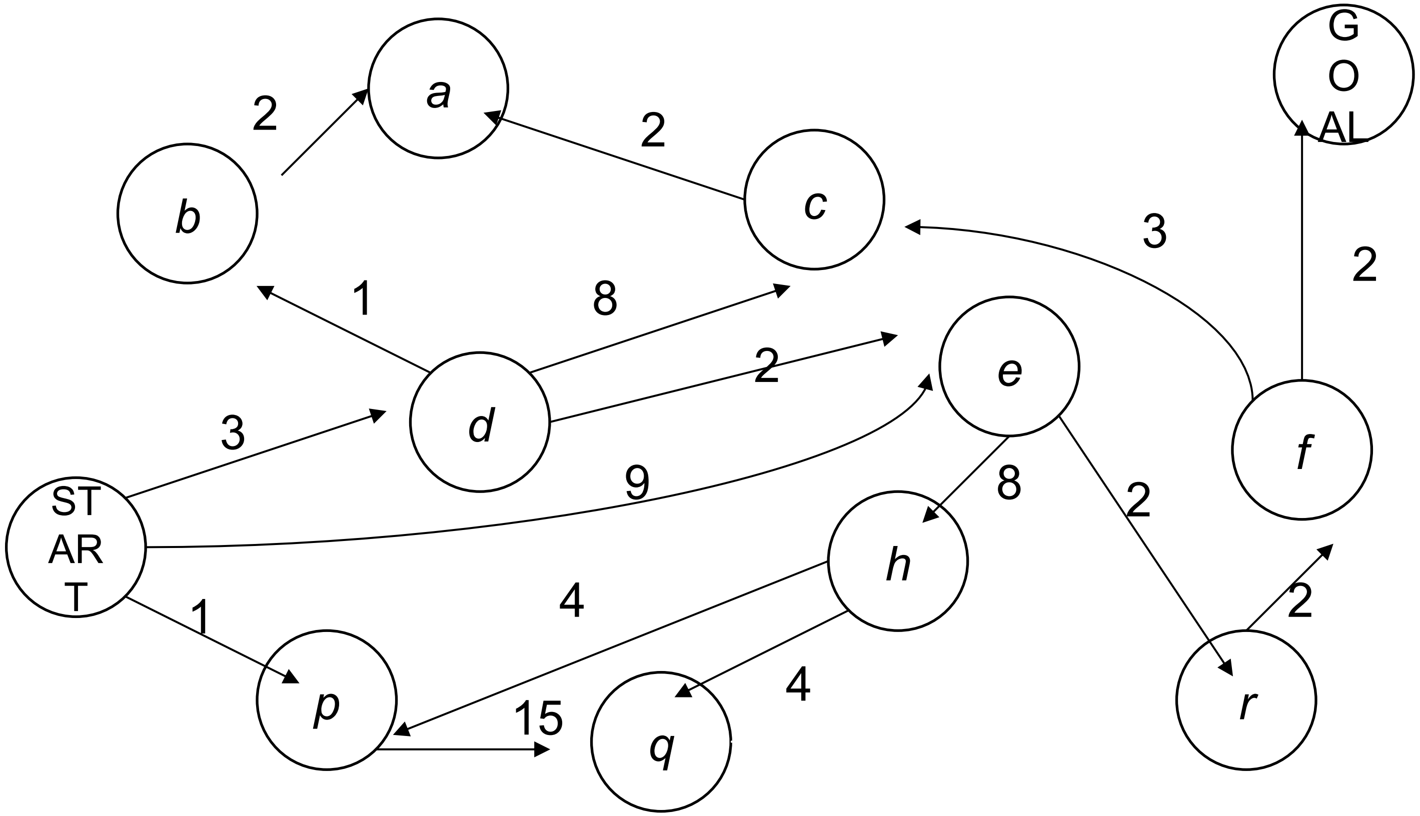
- Usefull when you know the **diameter** of the state-space graph

Iterative Deepening Search (IDS)

- Idea: get DFS's space advantage with BFS's time / shallow-solution advantages
  - Run a DFS with depth limit 1. If no solution…
  - Run a DFS with depth limit 2. If no solution…
  - Run a DFS with depth limit 3. …..
- Time complexity?
  - Search takes time $O(b^d)$
- Space complexity?
  - $O(bd)$
- Isn't that wastefully redundant?
  - Generally most work happens in the lowest level searched, so not so bad!

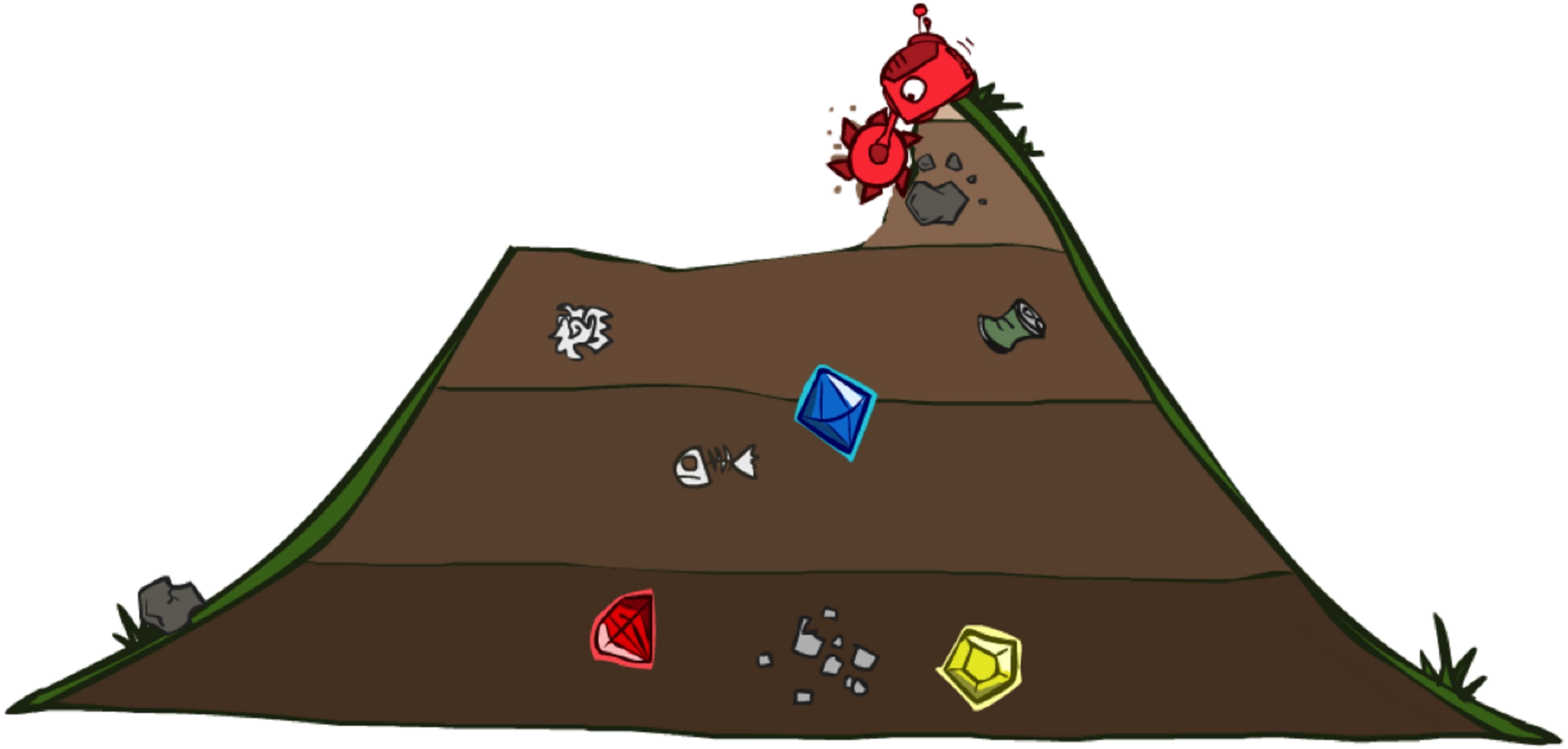


Cost-Sensitive Search



BFS finds the shortest path in terms of number of actions. It does not find the least-cost path. We will now cover a similar algorithm which does find the least-cost path.

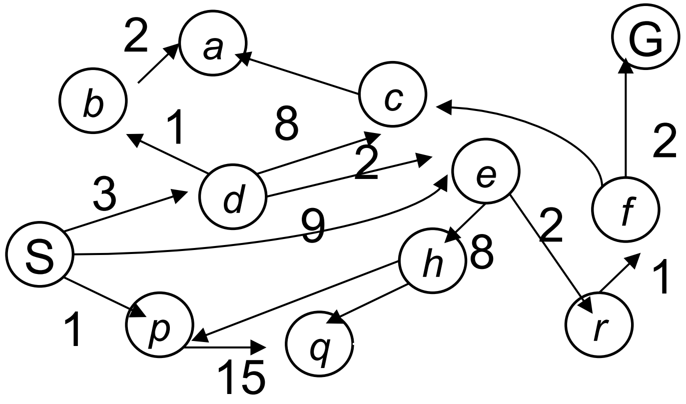
Uniform Cost Search (UCS) (Dijkstra's algorithm)

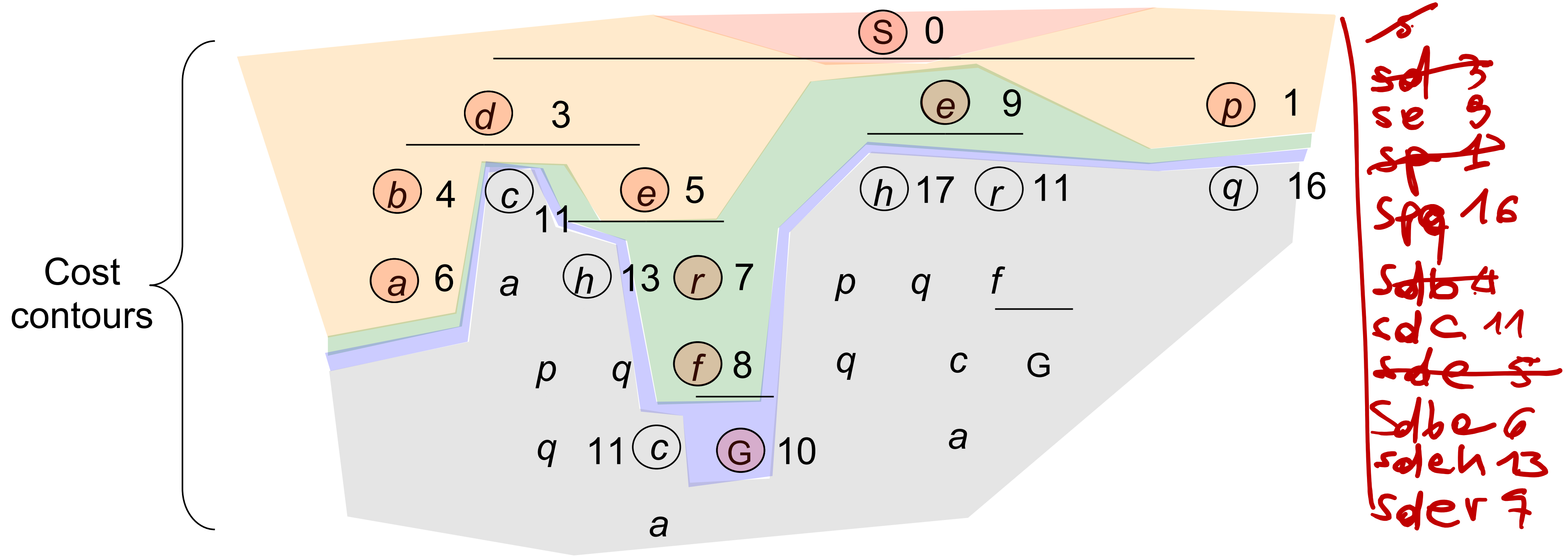


Uniform Cost Search

Strategy: expand a cheapest node first:

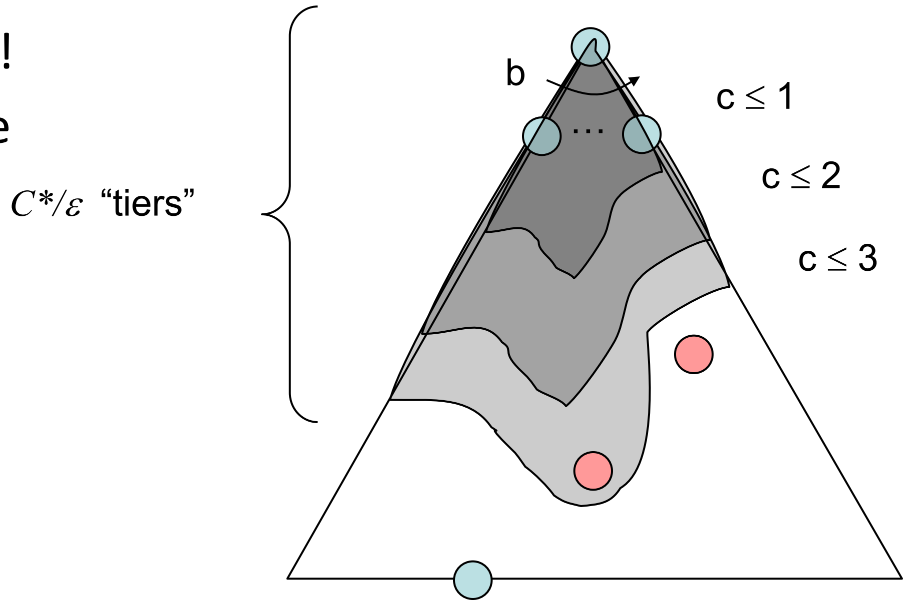
Fringe is a priority queue (priority: cumulative cost)





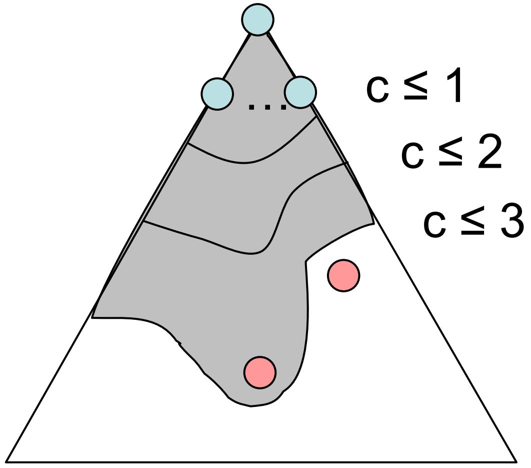
Uniform Cost Search (UCS) Properties

- What nodes does UCS expand?
  - Processes all nodes with cost less than cheapest solution!
  - If that solution costs C^* and arcs cost at least ε, then the "effective depth" is roughly C^*/ε
  - Takes time $O(b^{1+C^*/\varepsilon})$ (exponential in effective depth)
- How much space does the fringe take?
  - Has roughly the last tier, so $O(b^{1+C^*/\varepsilon})$
- Is it complete?
  - Assuming best solution has a finite cost and minimum arc cost is positive, yes!
- Is it optimal?
  - Yes! (Proof next lecture via A*)



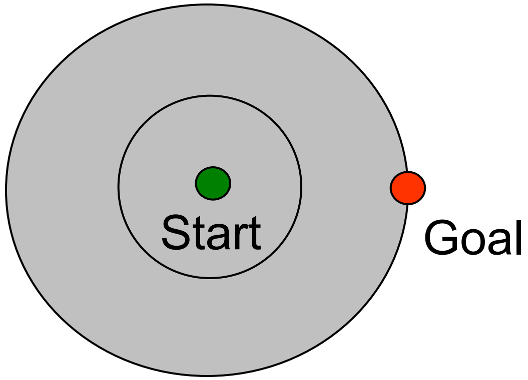
Uniform Cost Issues

- Remember: UCS explores increasing cost contours



- The good: UCS is complete and optimal!

- The bad:
  - Explores options in every "direction"
  - No information about goal location



- We'll fix that soon!

Video of Demo Contours UCS Pacman Small Maze

Video of Demo Empty UCS

Video of Demo Maze with Deep/Shallow Water --- DFS, BFS, or UCS? (part 1)

Video of Demo Maze with Deep/Shallow Water --- DFS, BFS, or UCS? (part 2)

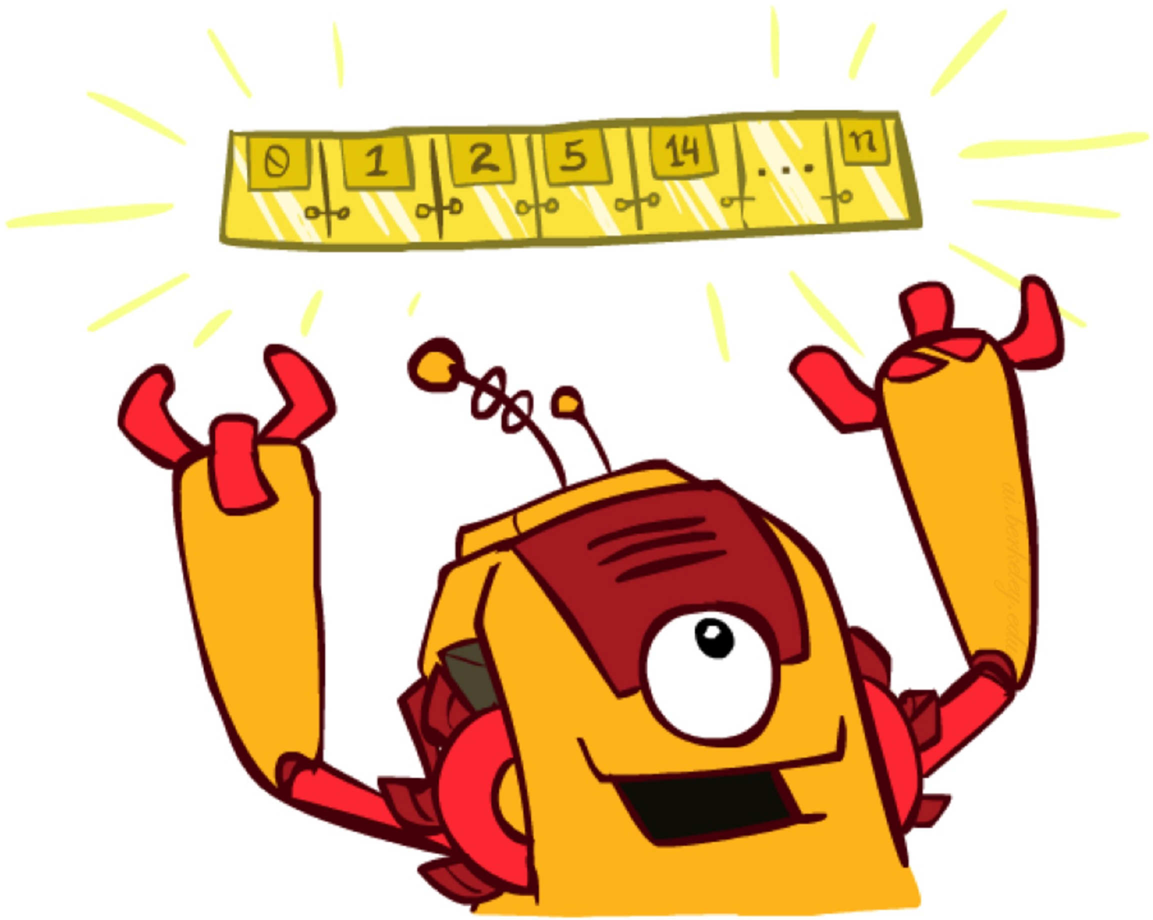
Video of Demo Maze with Deep/Shallow Water --- DFS, BFS, or UCS? (part 3)

Comparing uninformed search algorithms

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening
Complete?	Yes[1]	Yes[1,2]	No	No	Yes[1]
Optimal cost?	Yes[3]	Yes	No	No	Yes[3]
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$	$O(b^\ell)$	$O(b^d)$
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$	$O(b\ell)$	$O(bd)$

The One Queue

- All these search algorithms are the same except for fringe strategies
  - Conceptually, all fringes are priority queues (i.e. collections of nodes with attached priorities)
  - Practically, for DFS and BFS, you can avoid the log(n) overhead from an actual priority queue, by using stacks and queues
  - Can even code one implementation that takes a variable queuing object



Search and Models

- Search operates over models of the world
  - The agent doesn't actually try all the plans out in the real world!
  - Planning is all "in simulation"
  - Your search is only as good as your models…