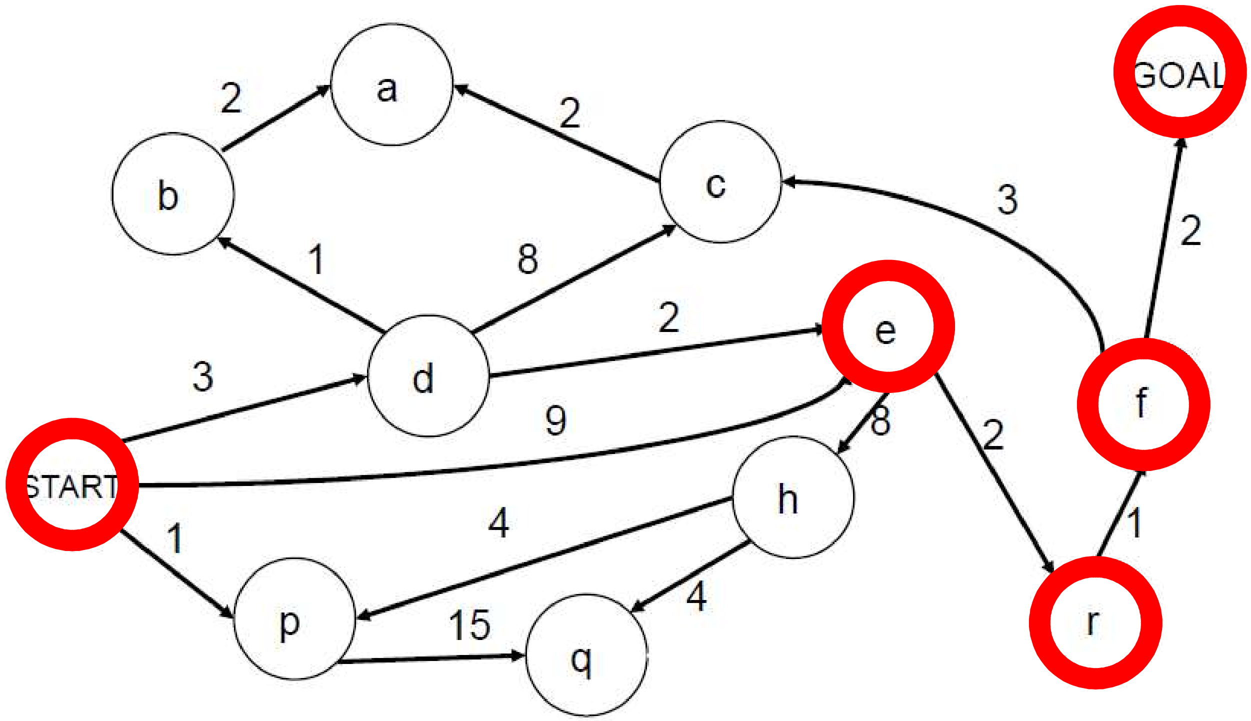
Search with varying step costs



- BFS finds the path with the fewest steps, but does not always find the cheapest path

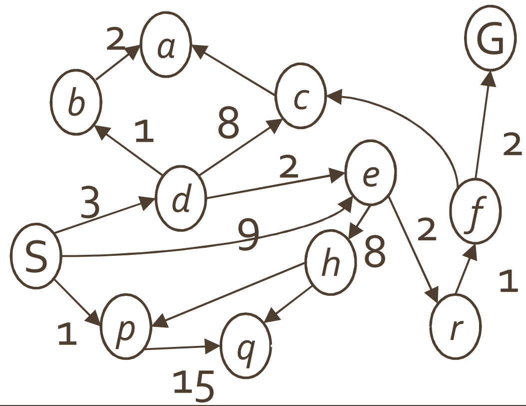
Uniform-cost search

- For each frontier node, save the total cost of the path from the initial state to that node
- Expand the frontier node with the lowest path cost
- Implementation: *frontier* is a priority queue ordered by path cost
- Equivalent to BFS if step costs all equal
- Equivalent to Dijkstra’s algorithm in general

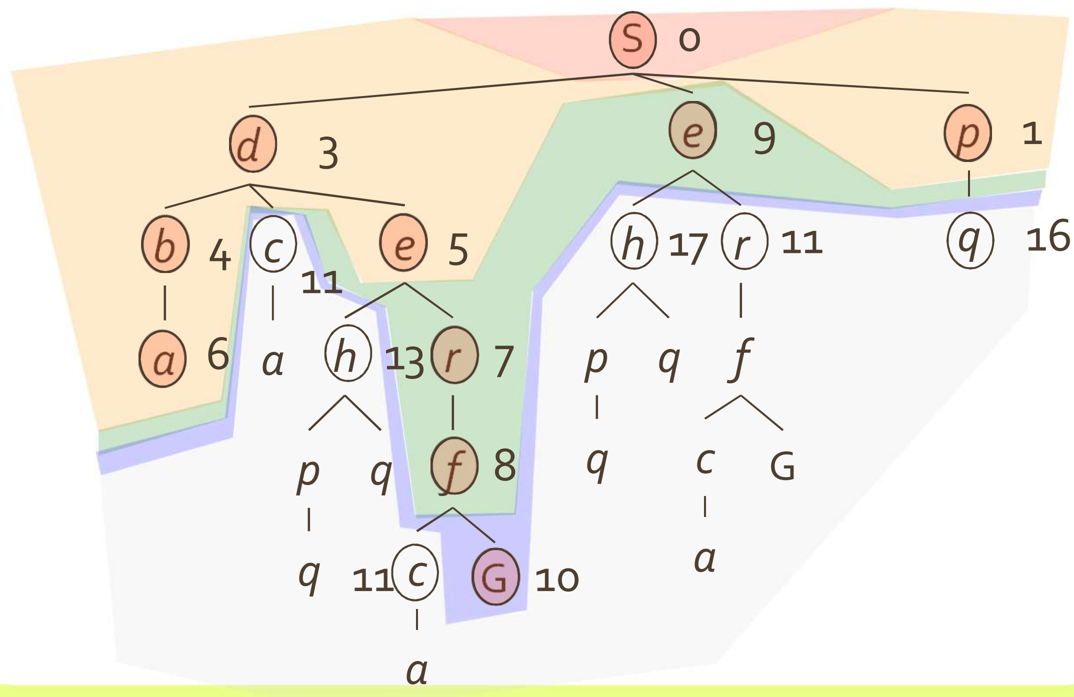
Uniform Cost Search

Strategy: expand a cheapest node first:

Frontier is a priority queue (priority: cumulative cost)

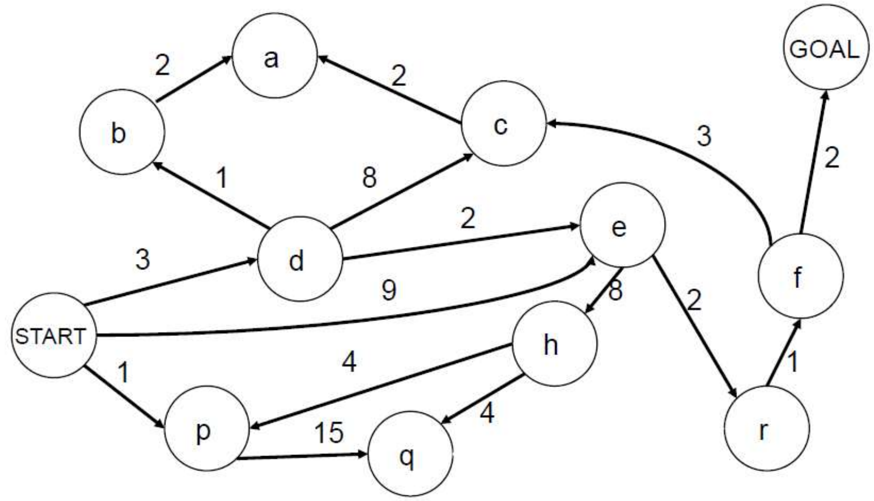


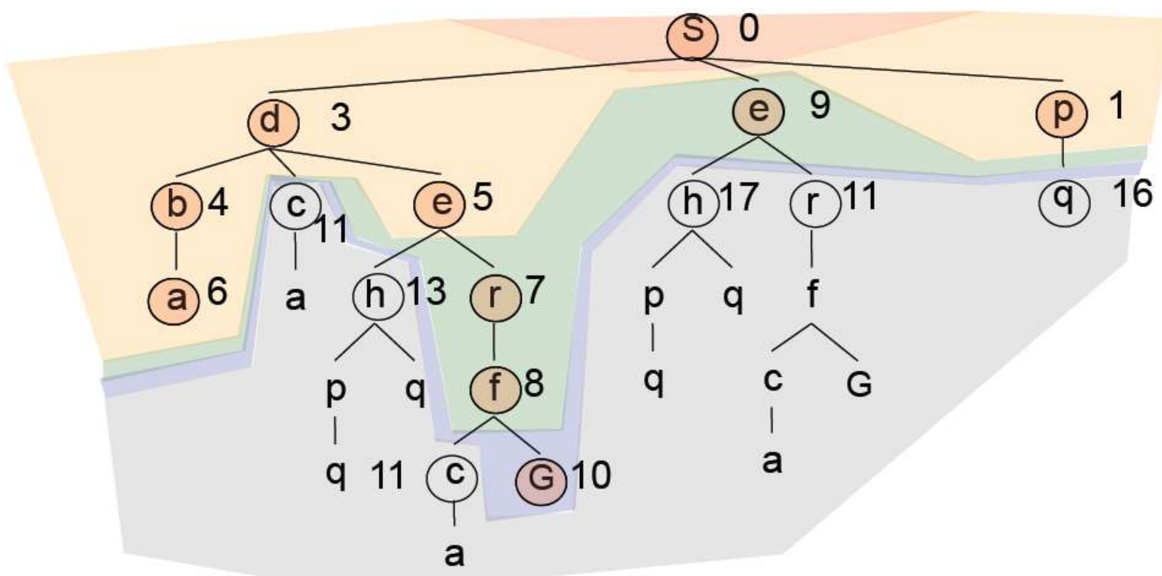
Cost contours



Uniform-cost search example

- Expansion order:
  (S,p,d,b,e,a,r,f,e,G)





Another example of uniform-cost search

Source: Wikipedia

Properties of uniform-cost search

- **Complete?**

  Yes, if step cost is greater than some positive constant ε (we don't want infinite sequences of steps that have a finite total cost)

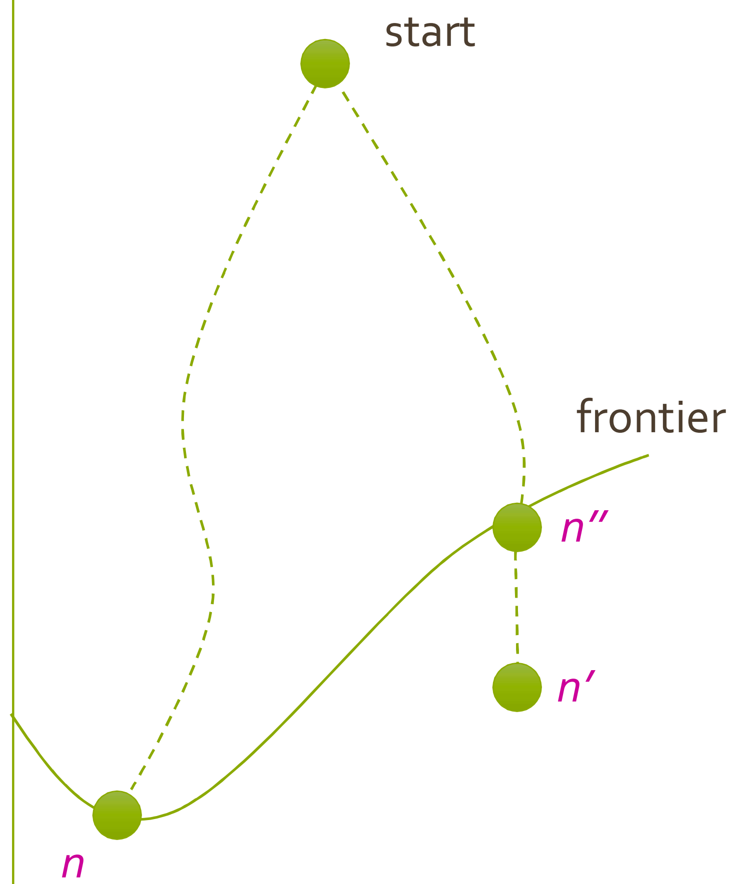
- **Optimal?**

  Yes

Optimality of uniform-cost search

- **Graph separation property**: every path from the initial state to an unexplored state has to pass through a state on the frontier
  - Proved inductively

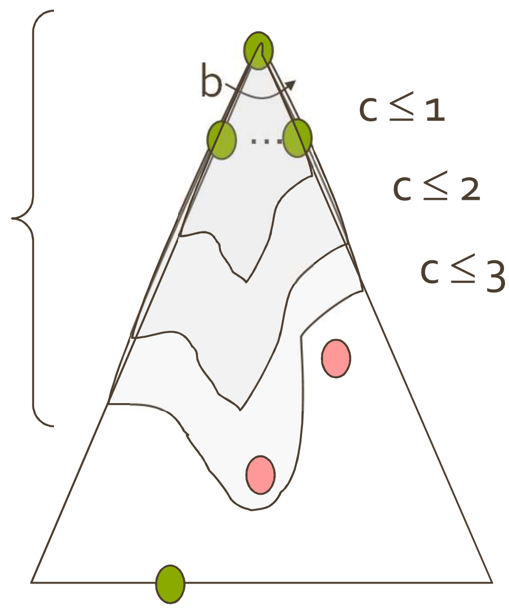
- *Optimality of UCS: proof by contradiction*
  - *Suppose UCS terminates at goal state n with path cost g(n) but there exists another goal state n′ with g(n′) < g(n)*
  - *By the graph separation property, there must exist a node n″ on the frontier that is on the optimal path to n′*
  - *But because g(n″) ≤ g(n′) < g(n), n″ should have been expanded first!*



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$^{C^*/\varepsilon}$) (exponential in effective depth)

C^*/ε "tiers"



- How much space does the frontier take?
  - Has roughly the last tier, so O(b$^{C^*/\varepsilon}$)

- Is it complete?
  - Assuming best solution has a finite cost and minimum arc cost is positive, yes!

- Is it optimal?
  - Yes!

Properties of uniform-cost search

- **Complete?**

  Yes, if step cost is greater than some positive constant ε (we don't want infinite sequences of steps that have a finite total cost)

- **Optimal?**

  Yes – nodes expanded in increasing order of path cost

- **Time?**

  Number of nodes with path cost ≤ cost of optimal solution (C^*), $O(b^{C^*/\varepsilon})$

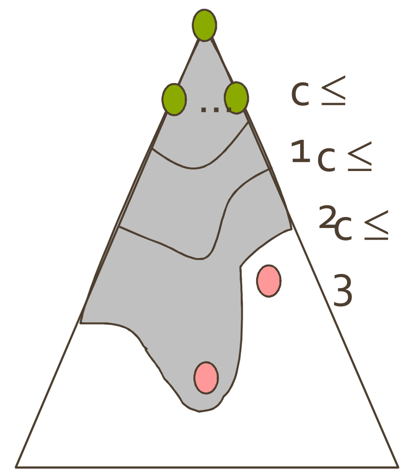
  This can be ***greater*** than $O(b^d)$: the search can explore long paths consisting of small steps before exploring shorter paths consisting of larger steps

- **Space?**

  $O(b^{C^*/\varepsilon})$

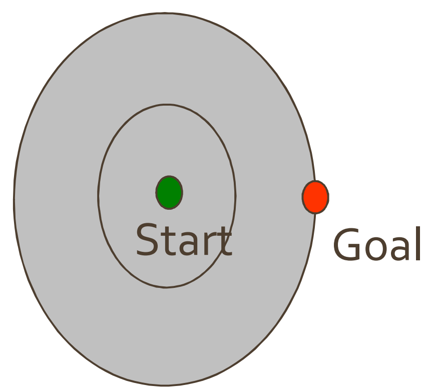
Uniform Cost Issues

- 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 Empty UCS

Review: Uninformed search strategies

Algorithm	Complete?	Optimal?	Time complexity	Space complexity
BFS	Yes	If all step costs are equal	$O(b^d)$	$O(b^d)$
DFS	No	No	$O(b^m)$	$O(bm)$
IDS	Yes	If all step costs are equal	$O(b^d)$	$O(bd)$
UCS	Yes	Yes	Number of nodes with $g(n) \le C^*$	

b: maximum branching factor of the search tree
d: depth of the optimal solution
m: maximum length of any path in the state space
C*: cost of optimal solution
g(n): cost of path from start state to node n