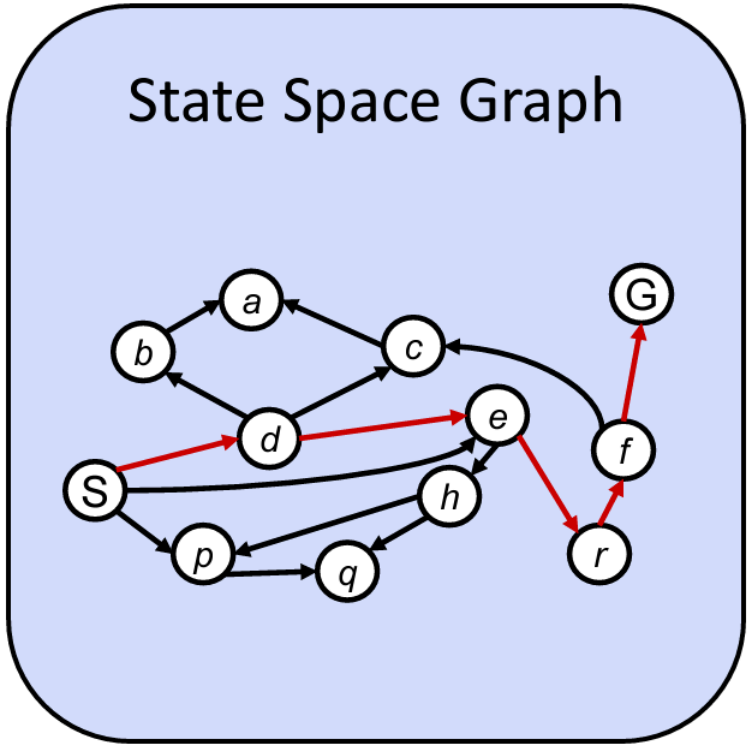
ME 620: Fundamentals of Artificial Intelligence

Lecture 6: Uninformed Search - II

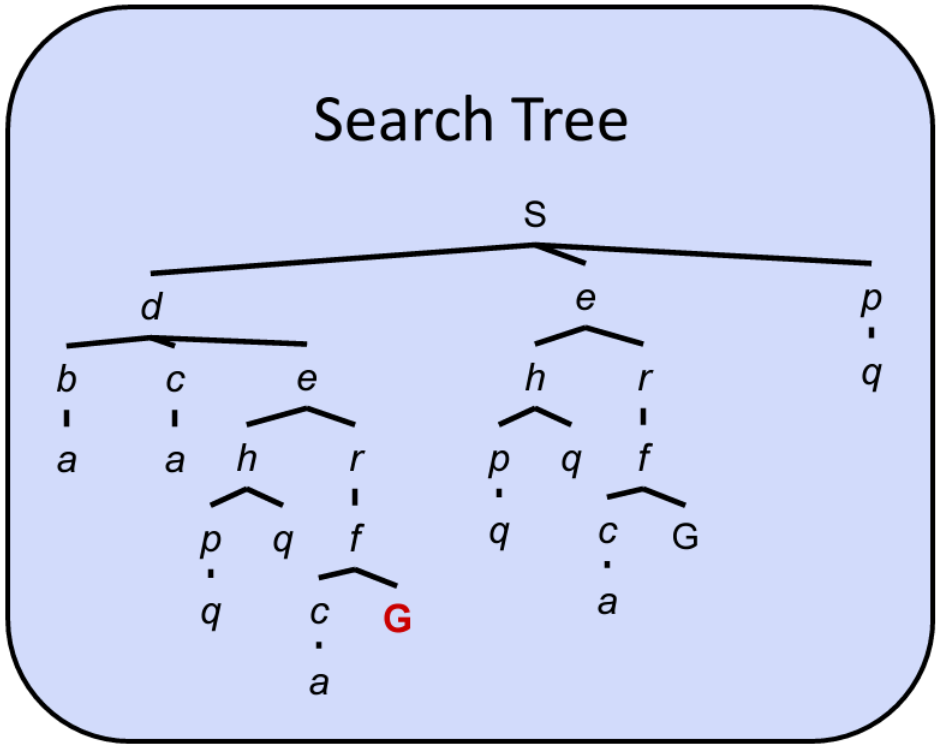
Shyamanta M Hazarika
Biomimetic Robotics and Artificial Intelligence Lab
Mechanical Engineering and M F School of Data Sc. & AI
IIT Guwahati

State Space Graph Vs. Search Tree



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

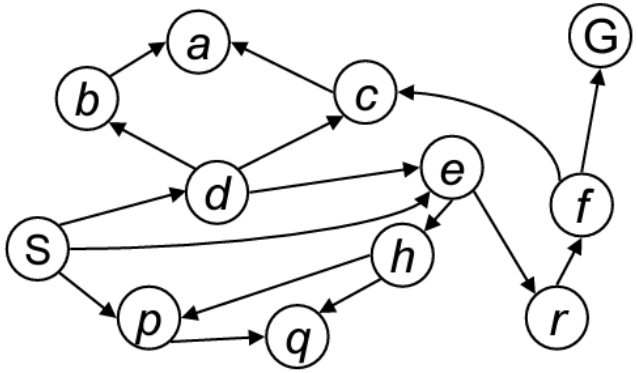
We construct both on demand – and we construct as little as possible.

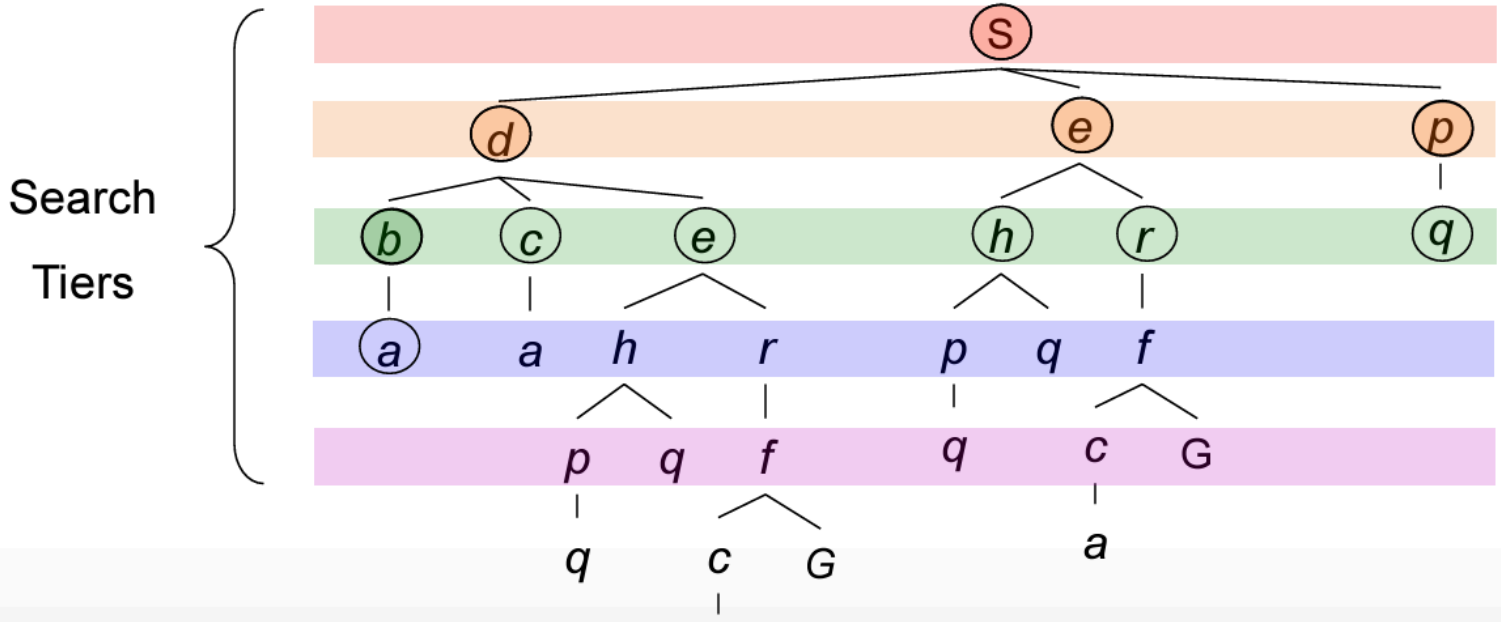


Breadth First Search

Strategy: expand a shallowest node first

Implementation: Fringe is a FIFO queue

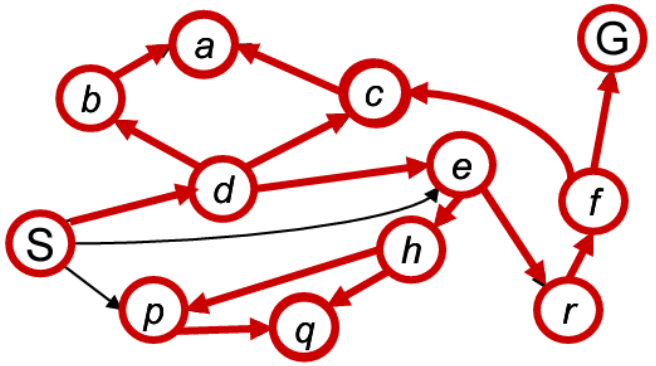


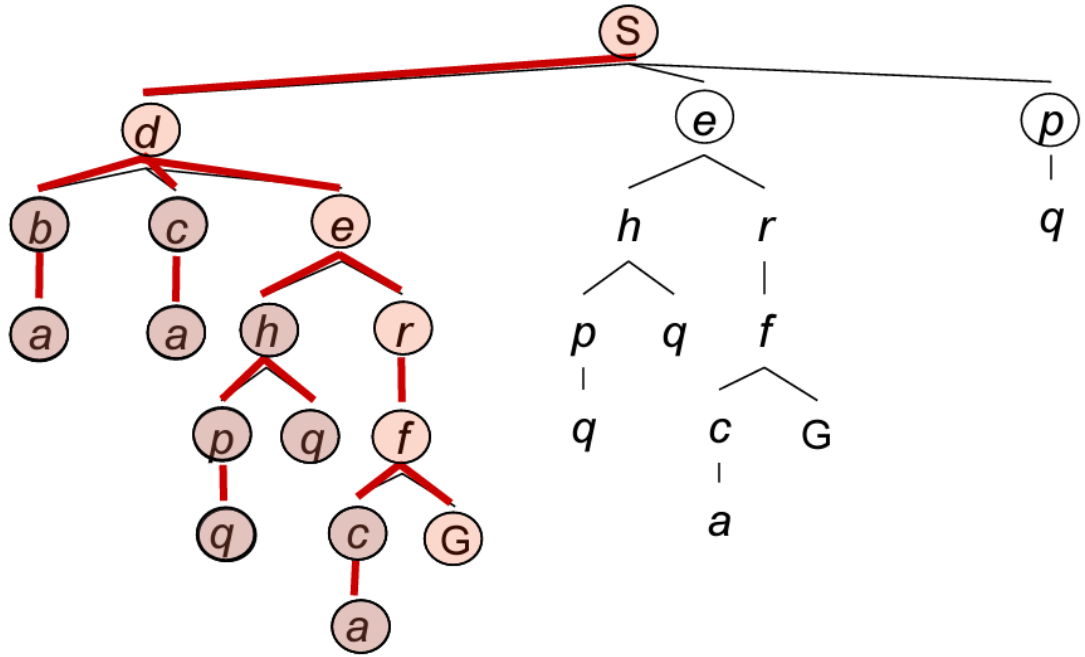


Depth First Search

Strategy: expand a deepest node first

Implementation: Fringe is a LIFO stack





Iterative deepening search

Depth-Limited Search

To avoid the infinite depth problem of DFS, we can decide to only search until depth L, i.e. we don't expand beyond depth L.

Iterative Deepening Search

What of solution is deeper than L?

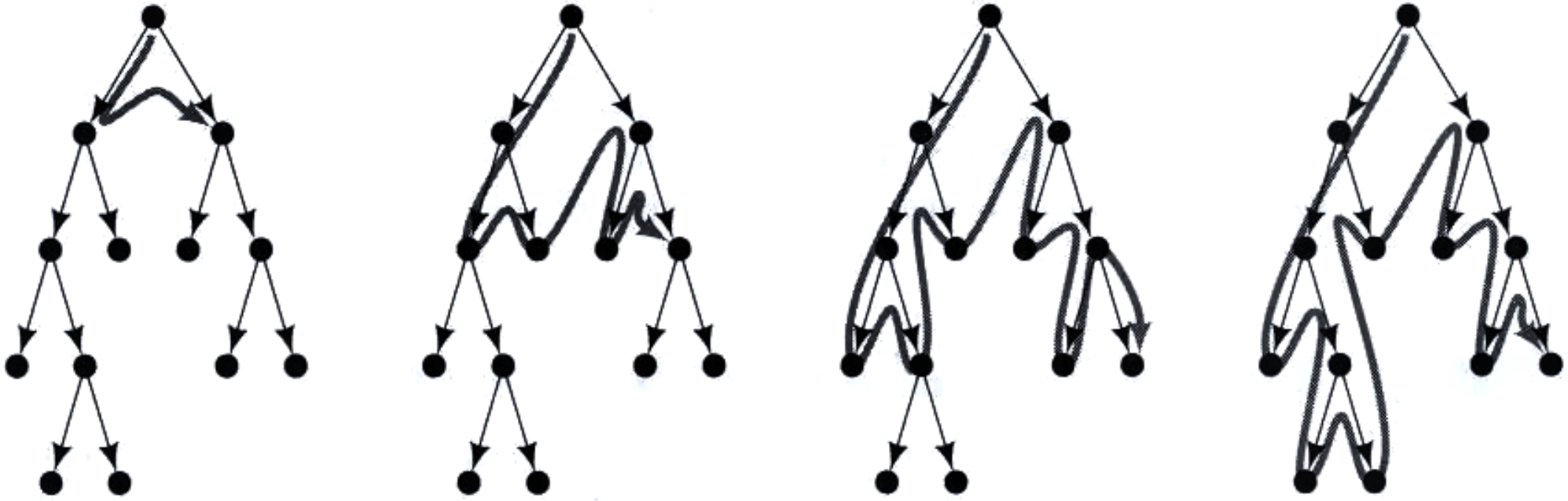
Increase L iteratively.

As we shall see, Iterative Deepening Search inherits the memory advantage of Depth-First search.

Iterative deepening search

- 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. …..

Example IDS



Stages in Iterative-Deepening Search

Properties of Iterative Deepening search

- Complete?
  - Yes
- Time?
  - $(d+1)b^0 + d\,b^1 + (d-1)b^2 + \ldots + b^d = O(b^d)$

Time complexity is a little worse than BFS or DFS because nodes near the top of the search tree are generated multiple times, but because almost all of the nodes are near the bottom of a tree, the worst case time complexity is still exponential, O(b^d)

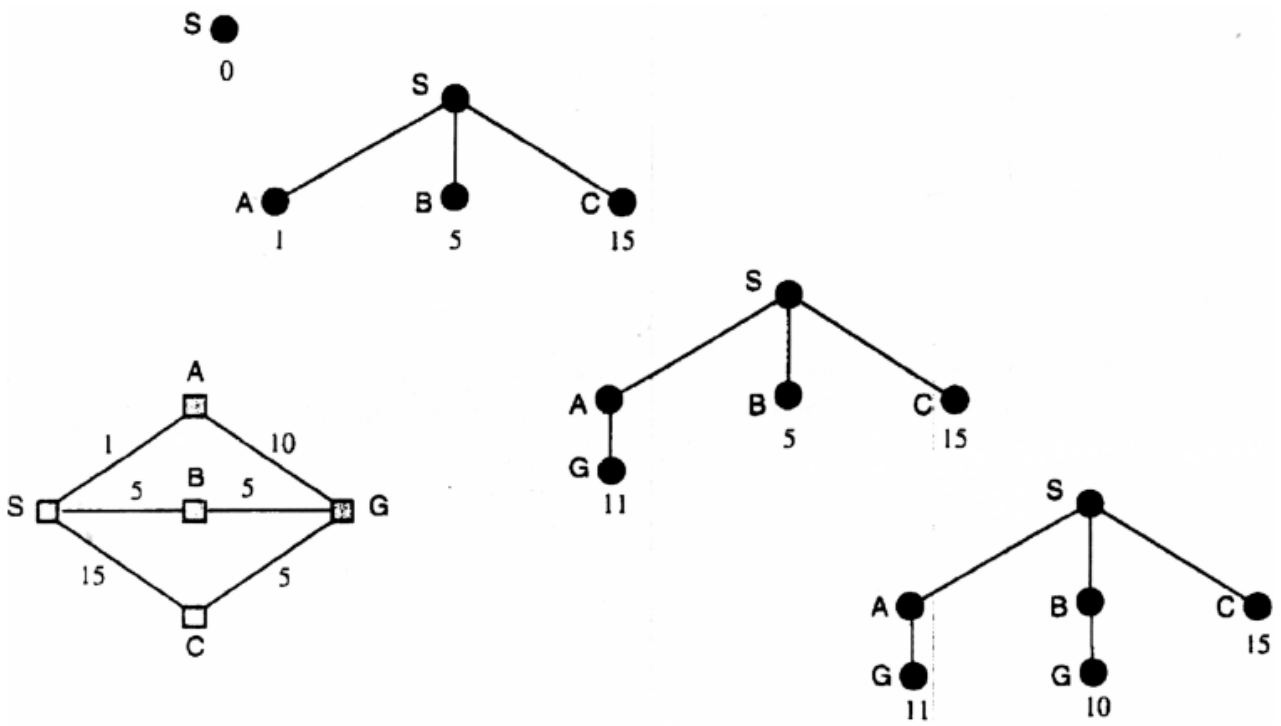
Example: If branching factor is b and solution is at depth d, then all nodes at depth d are generated at most once, all nodes at depth d-1 are generated at most twice, etc. Hence b^d + 2b^(d-1) + ... + db <= b^d / (1 - 1/b)^2 = O(b^d).

Properties of Iterative Deepening search

- Complete?
  - Yes
- Time?
  - $(d+1)b^0 + d\,b^1 + (d-1)b^2 + \ldots + b^d = O(b^d)$
- Space?
  - $O(d)$
- Optimal?
  - Yes, if step cost = 1 or increasing function of depth.

Uniform-cost search

Breadth-first is only optimal if step costs is increasing with depth (e.g. constant). Can we guarantee optimality for any step cost? Uniform-cost Search: Expand node with smallest path cost g(n).



Uniform-cost search

Implementation: *fringe* = queue ordered by path cost
Equivalent to breadth-first if all step costs all equal.

Complete?

Yes, if step cost ≥ ε
(otherwise it can get stuck in infinite loops)

Time?

of nodes with *path cost* ≤ cost of optimal solution.

Space?

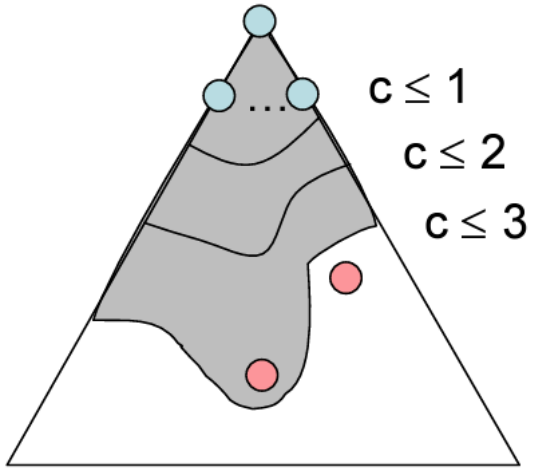
of nodes on paths with path cost ≤ cost of optimal solution.

Optimal?

Yes, for any step cost.

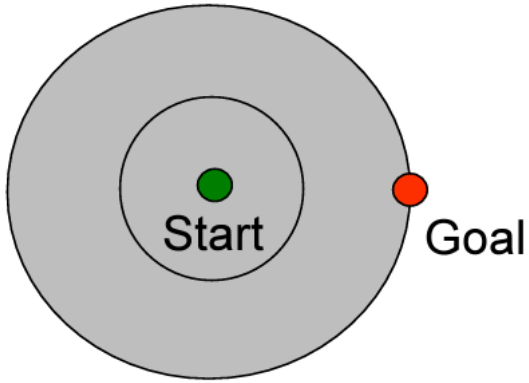
Uniform-cost search

- Remember: UCS explores increasing cost contours

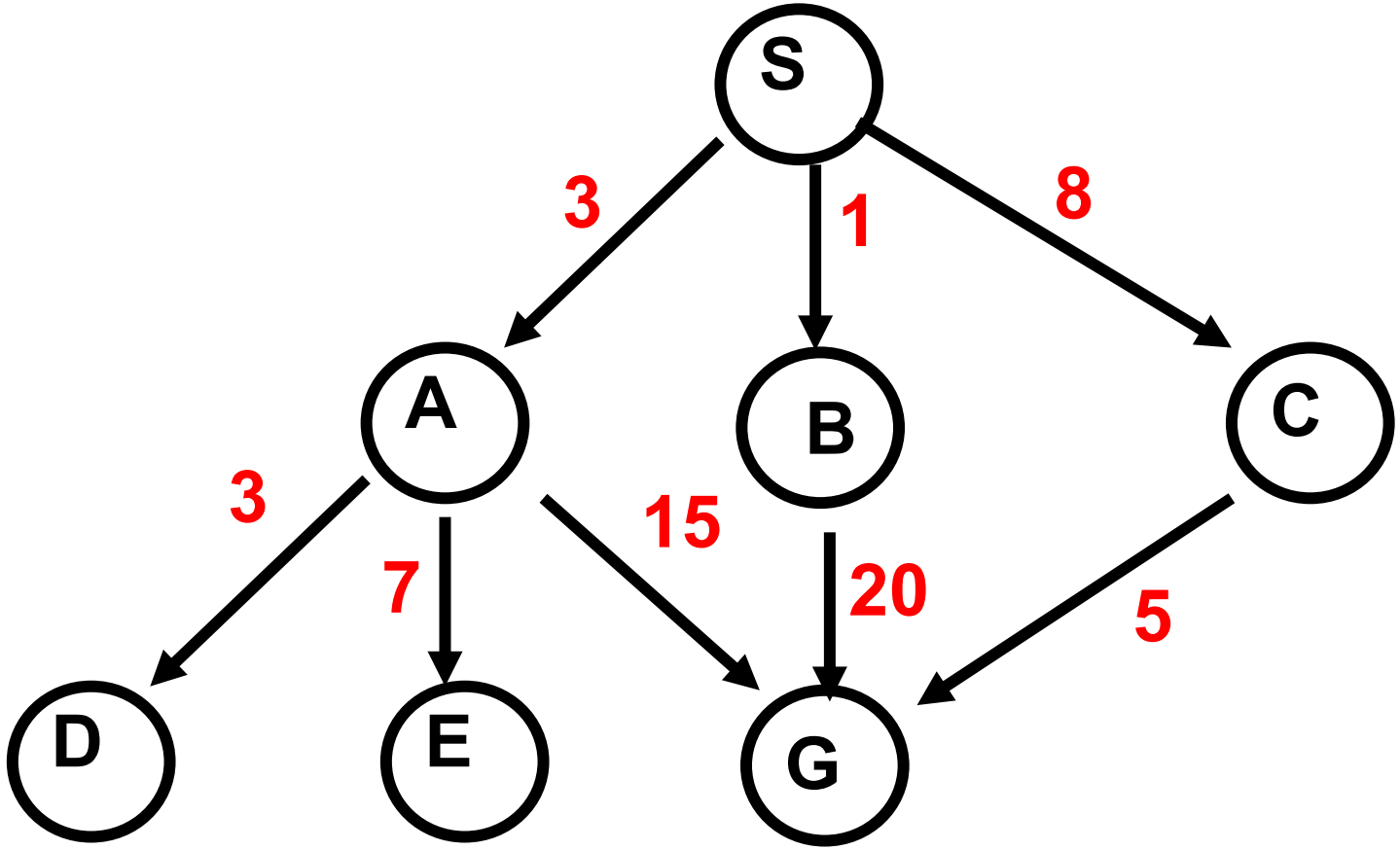


- The good: UCS is complete and optimal!

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

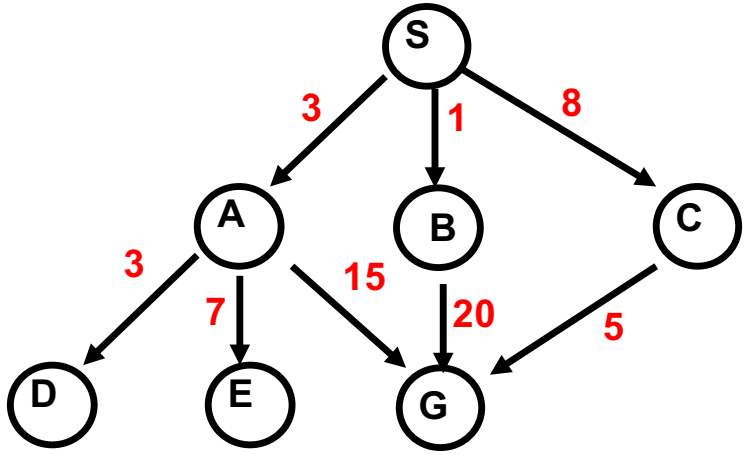


Illustrating Uninformed Search Strategies



Breadth-First Search

Expanded node	Nodes list
	{ S^0 }
S^0	{ A^3 B^1 C^8 }
A^3	{ B^1 C^8 D^6 E^{10} G^{18} }
B^1	{ C^8 D^6 E^{10} G^{18} G^{21} }
C^8	{ D^6 E^{10} G^{18} G^{21} G^{13} }
D^6	{ E^{10} G^{18} G^{21} G^{13} }
E^{10}	{ G^{18} G^{21} G^{13} }
G^{18}	{ G^{21} G^{13} }

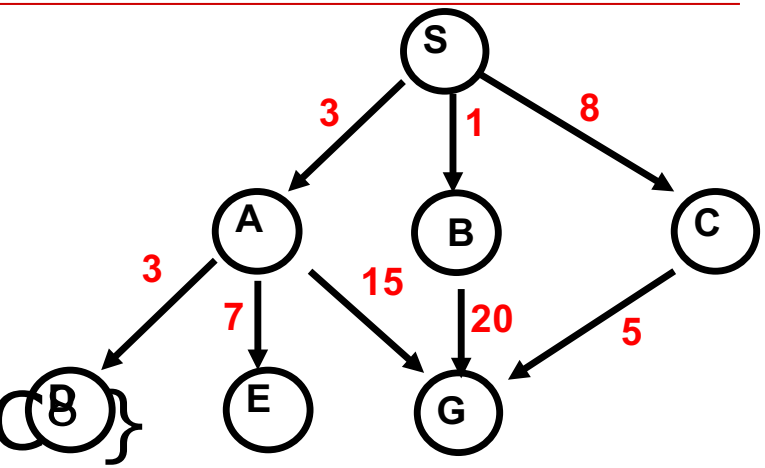


Solution path found is S A G , cost 18

Number of nodes expanded (including goal node) = 7

Depth-First Search

Expanded node	Nodes list
	{ S^0 }
S^0	{ A^3 B^1 C^8 }
A^3	{ D^6 E^{10} G^{18} B^1 C^8 }
D^6	{ E^{10} G^{18} B^1 C^8 }
E^{10}	{ G^{18} B^1 C^8 }
G^{18}	{ B^1 C^8 }

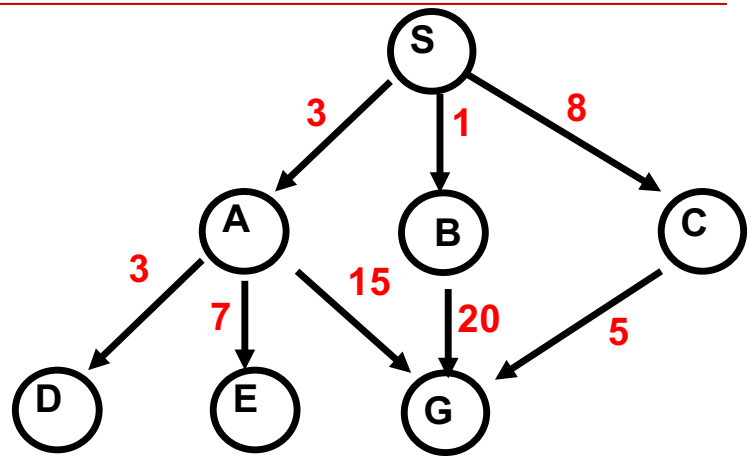


Solution path found is S A G, cost 18

Number of nodes expanded (including goal node) = 5

Uniform-Cost Search

Expanded node	Nodes list
	{ S^0 }
S^0	{ B^1 A^3 C^8 }
B^1	{ A^3 C^8 G^{21} }
A^3	{ D^6 C^8 E^{10} G^{18} G^{21} }
D^6	{ C^8 E^{10} G^{18} G^1 }
C^8	{ E^{10} G^{13} G^{18} G^{21} }
E^{10}	{ G^{13} G^{18} G^{21} }
G^{13}	{ G^{18} G^{21} }



Solution path found is S B G, cost 13

Number of nodes expanded (including goal node) = 7

How they perform

- **Breadth-First Search**:
  - Expanded nodes: S A B C D E G
  - Solution found: S A G (cost 18)
- **Depth-First Search:**
  - Expanded nodes: S A D E G
  - Solution found: S A G (cost 18)
- **Uniform-Cost Search**:
  - Expanded nodes: S A D B C E G
  - Solution found: S B G (cost 13)

  This is the only uninformed search that worries about costs.
- **Iterative-Deepening Search**:
  - Nodes expanded: S S A B C S A D E G
  - Solution found: S A G (cost 18)