Uninformed Search

Uninformed Search Strategies

- **Uninformed search**: strategies that order nodes without using any domain specific information.

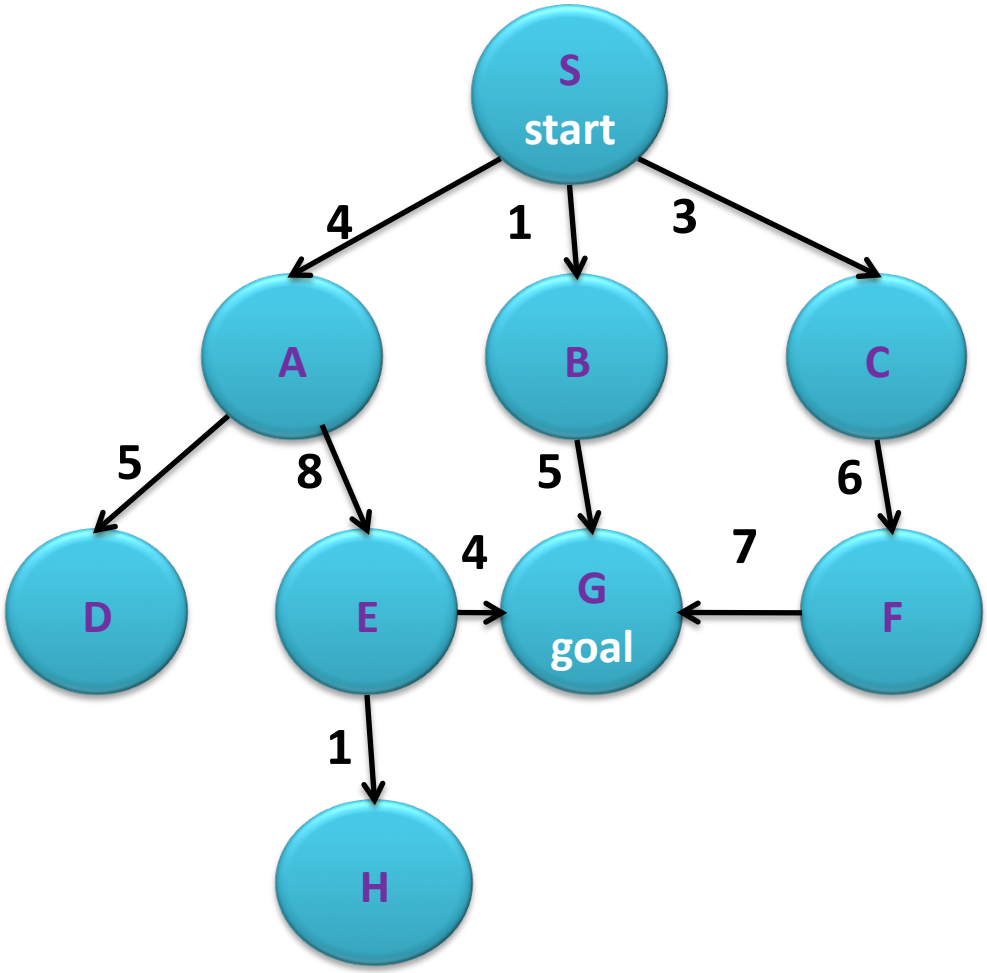
- BFS: **breadth-first search**
  - *queue (FIFO) used for the Fringe list
  - remove from front, add to back
- DFS: **depth-first search**
  - * stack (LIFO) used for the Fringe list
  - remove from front, add to front

Breadth-First Search(BFS)

General Search(Problem: queue)

#of nodes tested:0 expanded:0

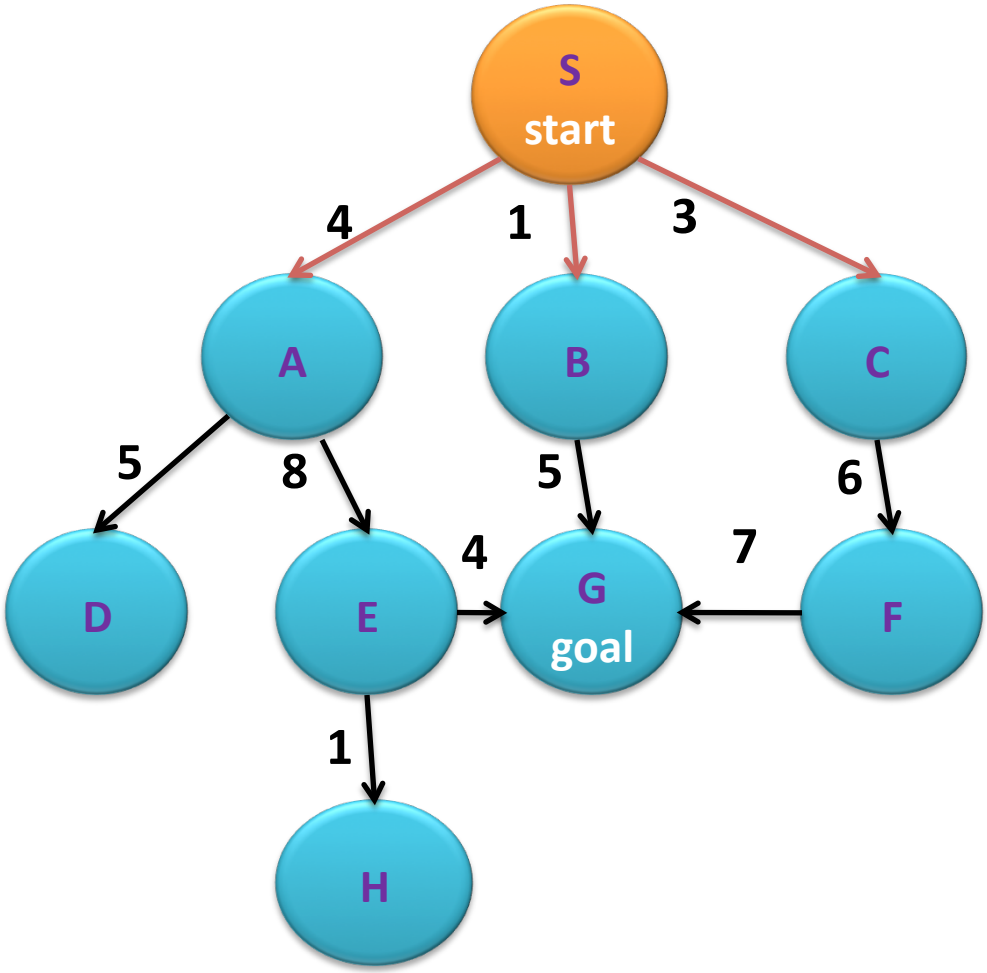
Expnd. node	Fringe list
	{S}



Breadth-First Search(BFS)

General Search(Problem: queue)

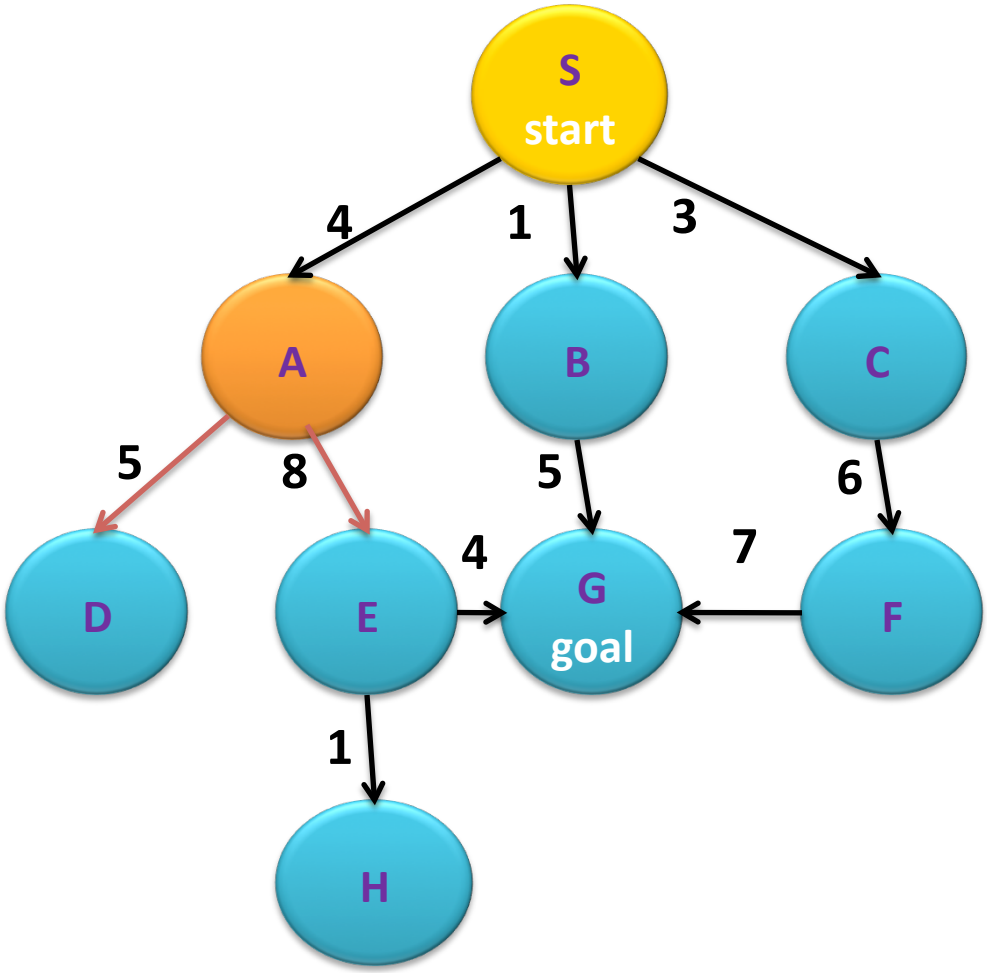
Expnd. node	Fringe list
	{S}
S not goal	{A,B,C}



Breadth-First Search(BFS)

General Search(Problem: queue)

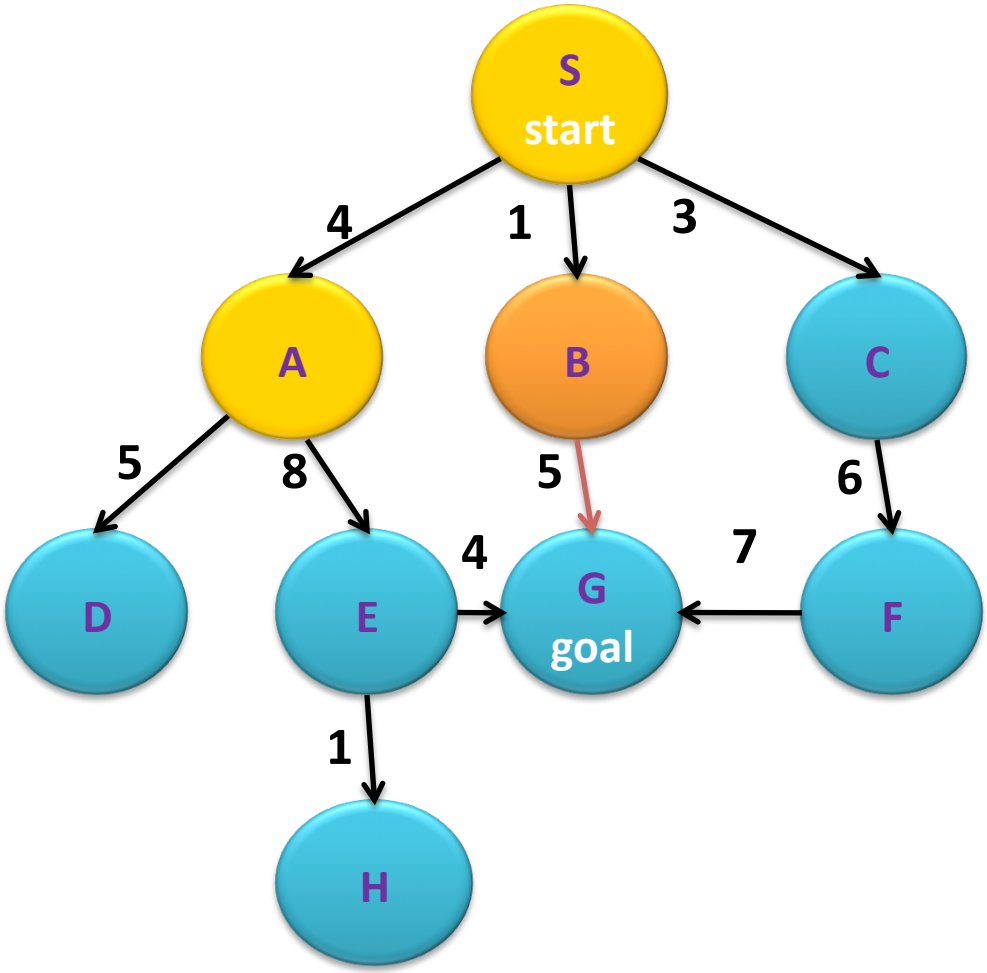
Expnd. node	Fringe list
	{S}
S	{A,B,C}
A not goal	{B,C,D, E}



Breadth-First Search(BFS)

General Search(Problem: queue)

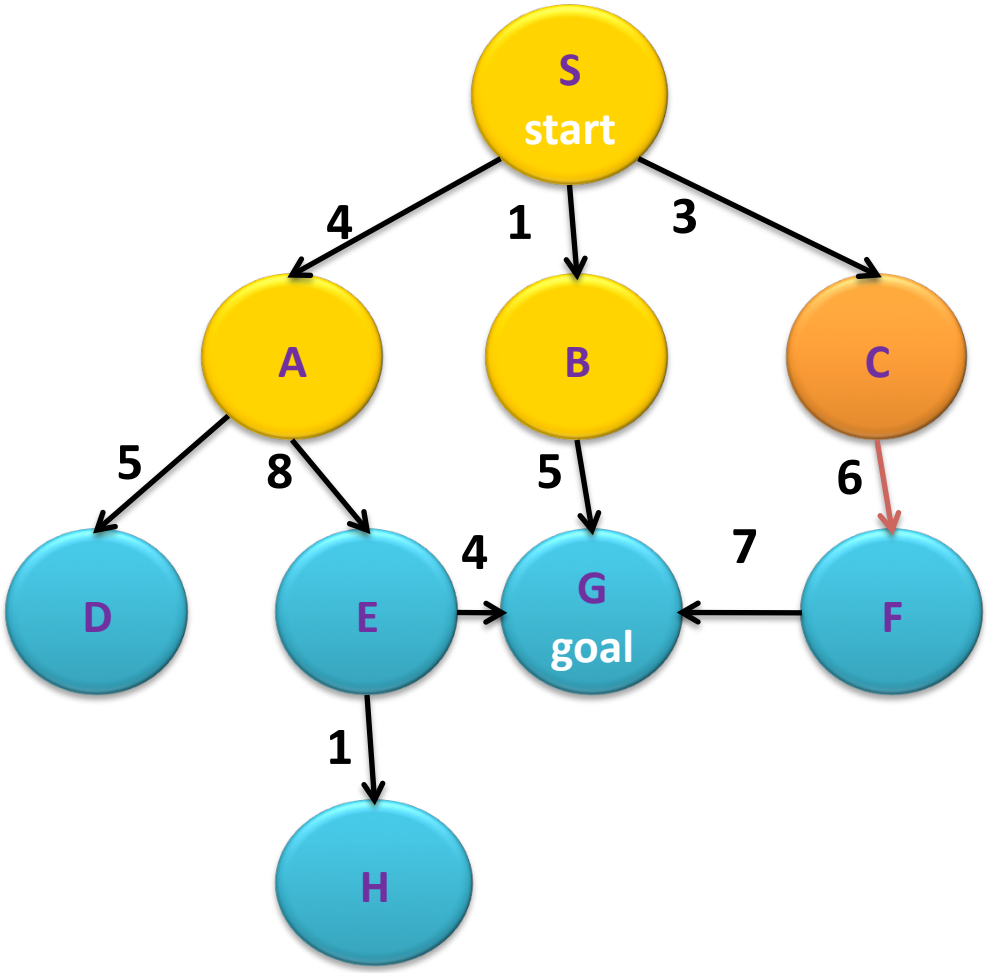
Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{B,C,D}
B not goal	{C,D,E,G}



Breadth-First Search(BFS)

General Search(Problem: queue)

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{B,C,D}
B	{C,D,E,G}
C not goal	{D,E,G,F}

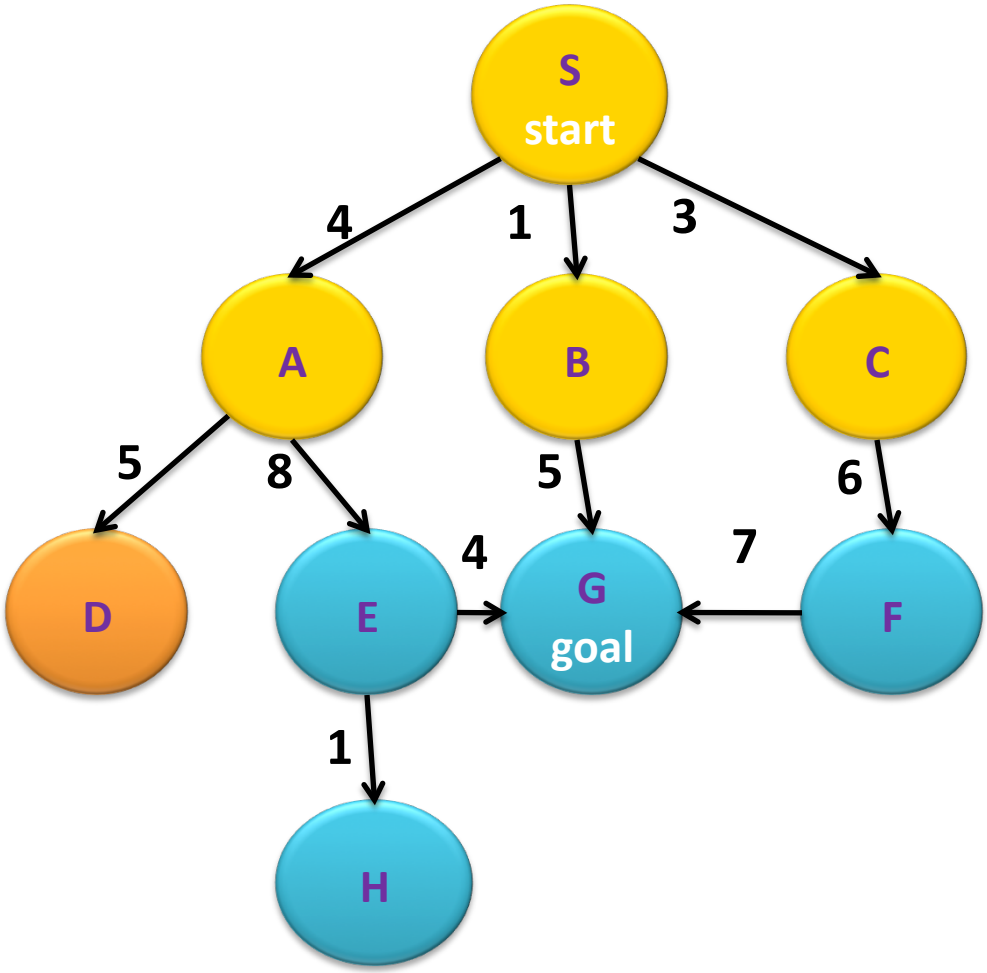


Breadth-First Search(BFS)

General Search

#of nodes tested:5 expanded:4

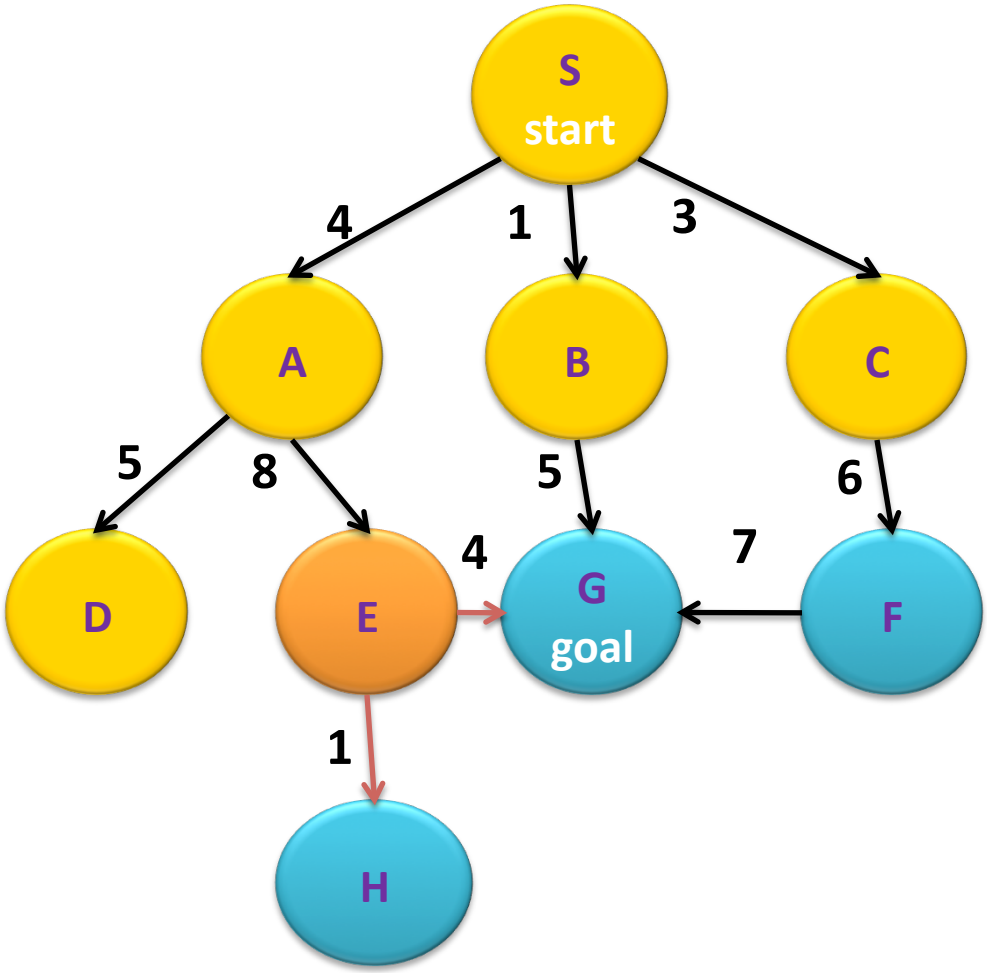
Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{B,C,D}
B	{C,D,E,G}
C	{D,E,G,F}
D not goal	{E,G,F} no expand



Breadth-First Search(BFS)

General Search(Problem: queue)

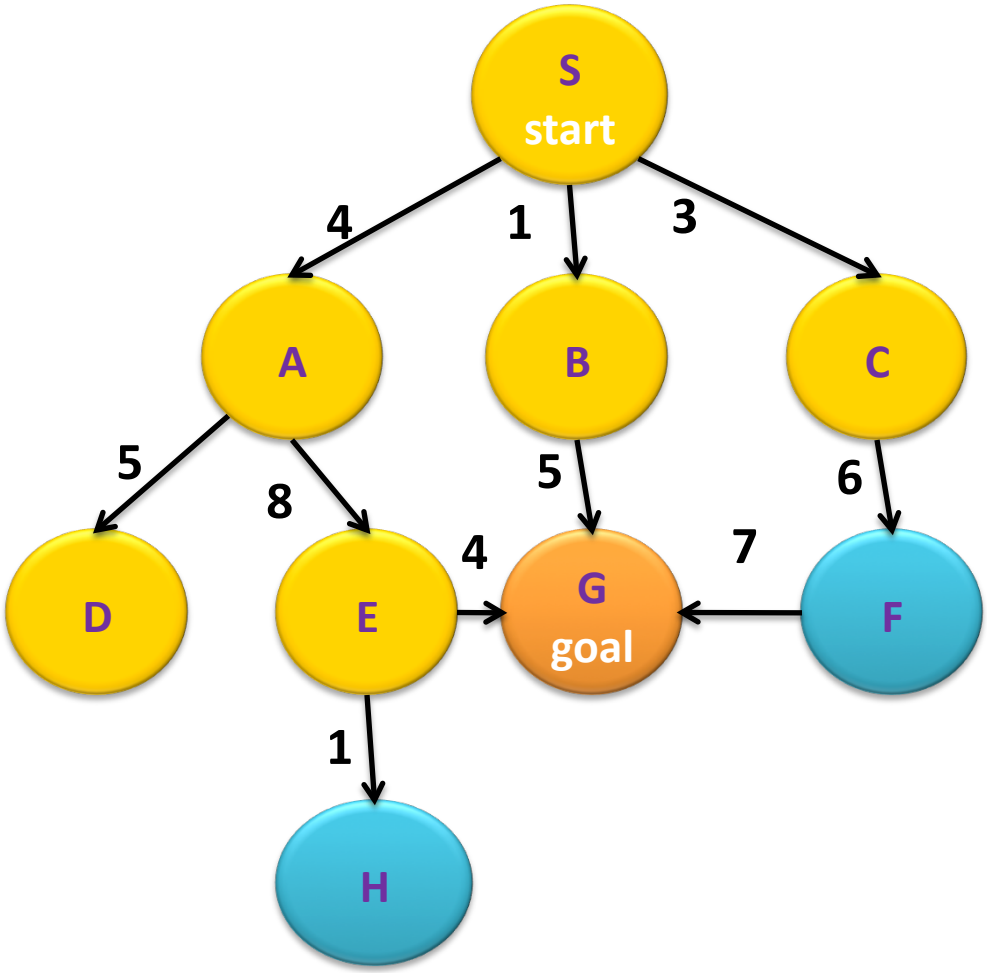
Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{B,C,D}
B	{C,D,E,G}
C	{D,E,G,F}
D	{E,G,F}
E not goal	{G,F,H,G}



Breadth-First Search(BFS)

General Search(Problem: queue)

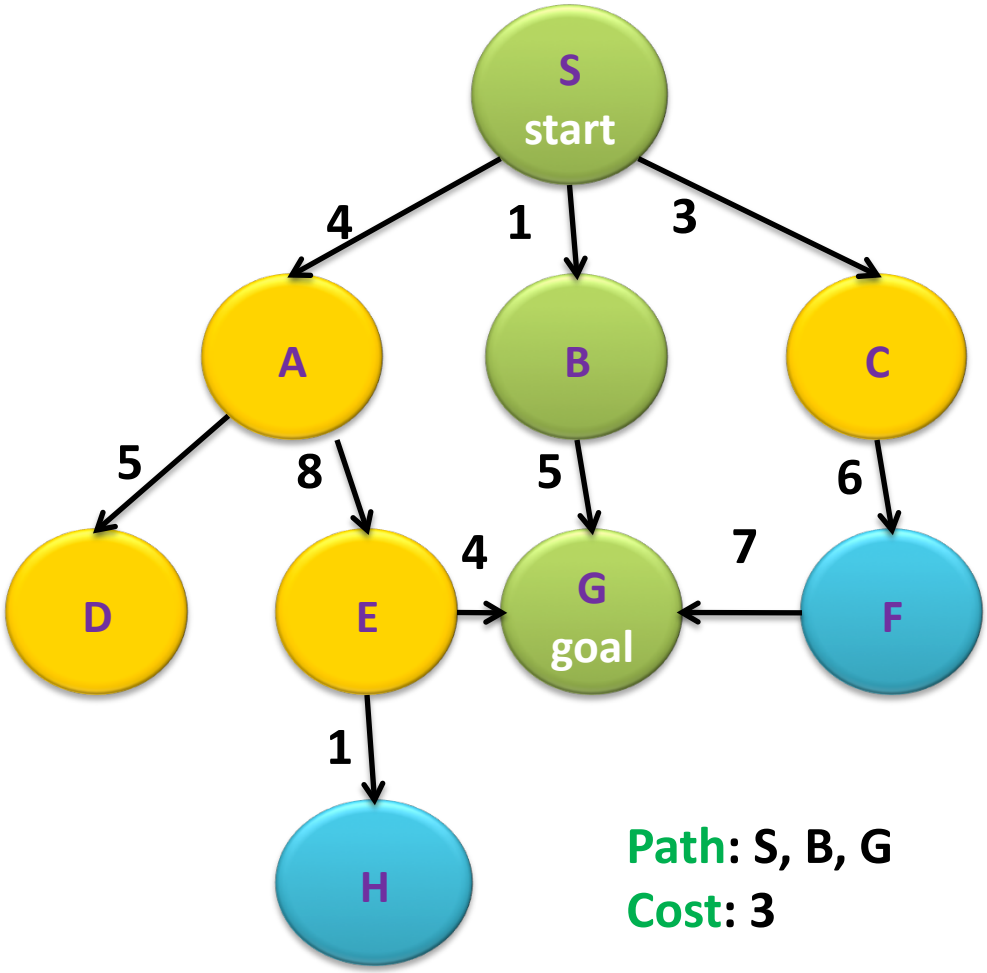
Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{B,C,D}
B	{C,D,E,G}
C	{D,E,G,F}
D	{E,G,F}
E	{G,F,H,G}
G goal	{F,H,G} no expand



Breadth-First Search(BFS)

General Search(Problem: queue)

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{B,C,D}
B	{C,D,E,G}
C	{D,E,G,F}
D	{E,G,F}
E	{G,F,H,G}
G	{F,H,G}



Breadth-First Search(BFS)

- Complete

- Optimal / Admissible
  - if all operators (i.e. arcs) have the same cost
  - otherwise, not optimal but does guarantee finding solution of shortest length (i.e., fewest arcs)

- Time and space complexity: $O(b^d)$ (i.e., exponential)
  - d is the depth of the solution
  - b is the branching factor at each non-leaf node

Breadth-First Search(BFS)

- A complete search tree has a total of nodes:

$$1 + b + b^2 + \ldots + b^d = (b^{(d+1)} - 1)/(b - 1)$$

  – d: the tree's depth

  – b: the branching factor at each non-leaf node

- For example: d=8, b=11

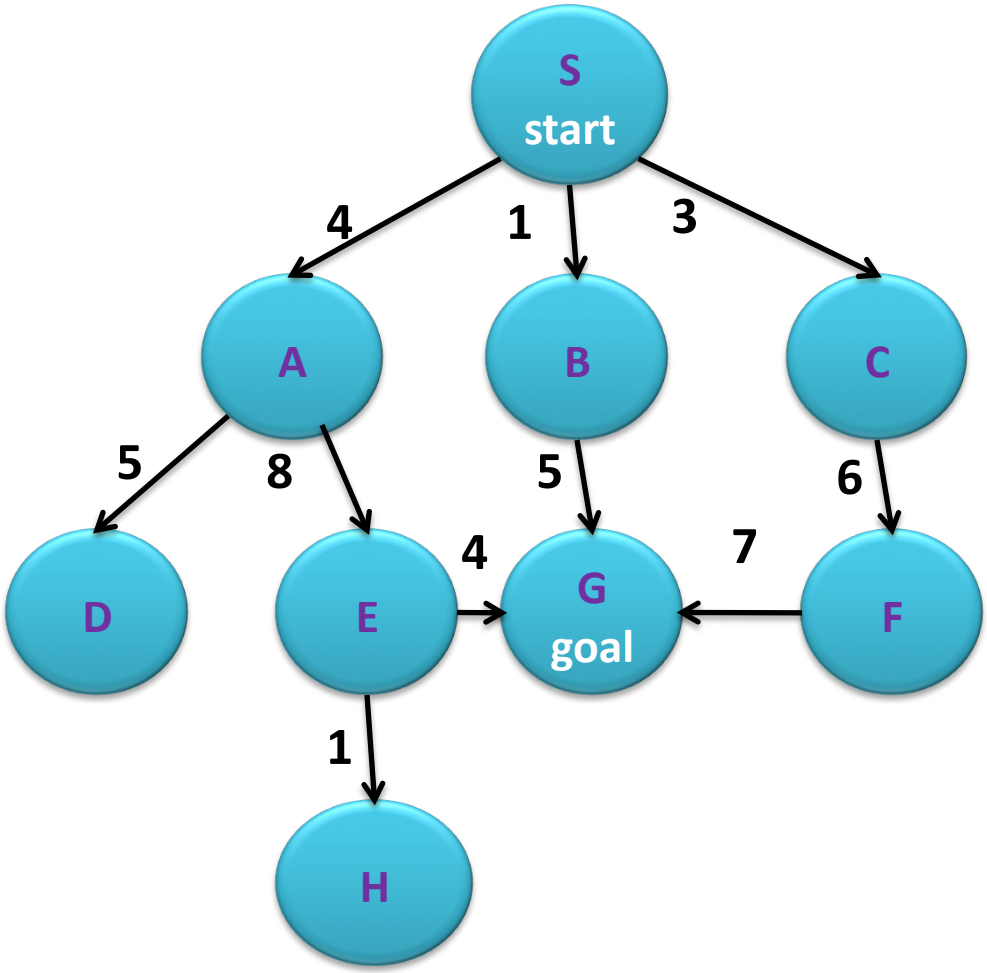
$$1 + 11 + 11^2 + \ldots + 11^8 = (11^9 - 1)/10 = O(11^8)$$

Depth-First Search(DFS)

General Search(Problem: stack)

#of nodes tested:0 expanded:0

Expnd. node	Fringe list
	{S}

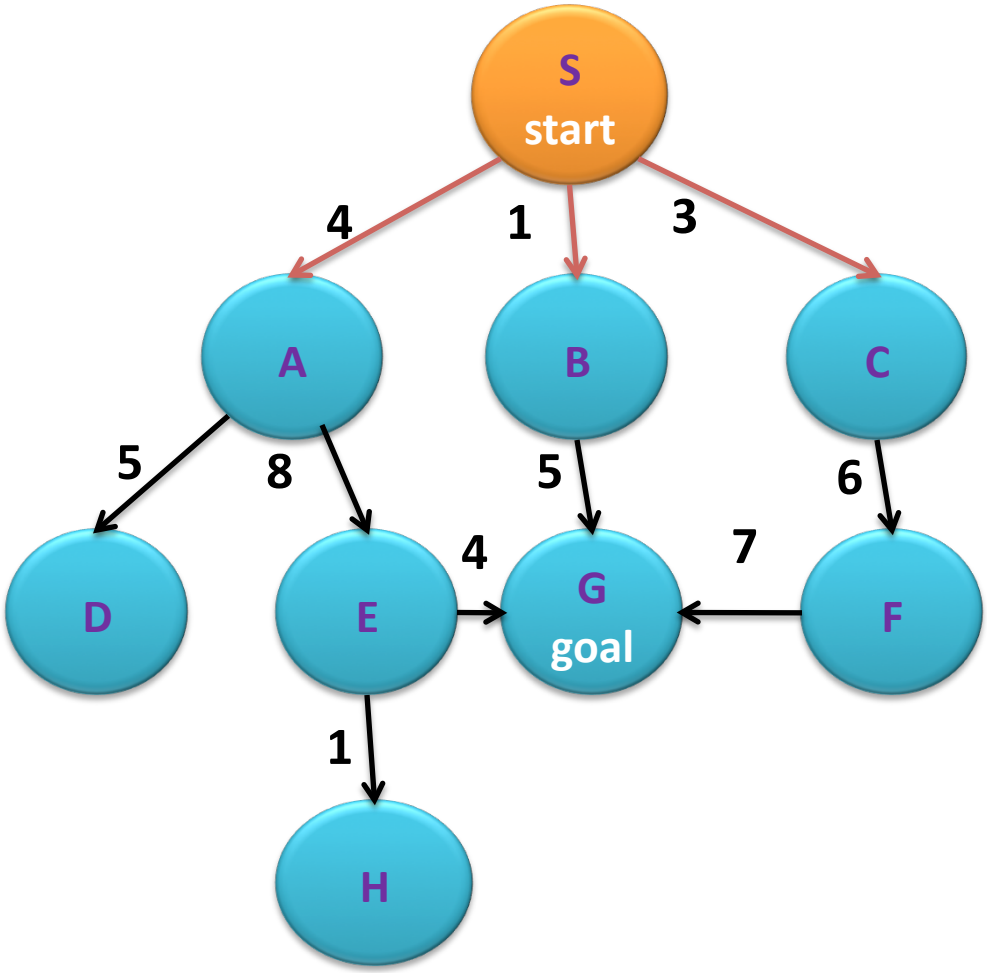


Depth-First Search(DFS)

General Search(Problem: stack)

#of nodes tested:1 expanded:1

Expnd. node	Fringe list
	{S}
S not goal	{A,B,C}

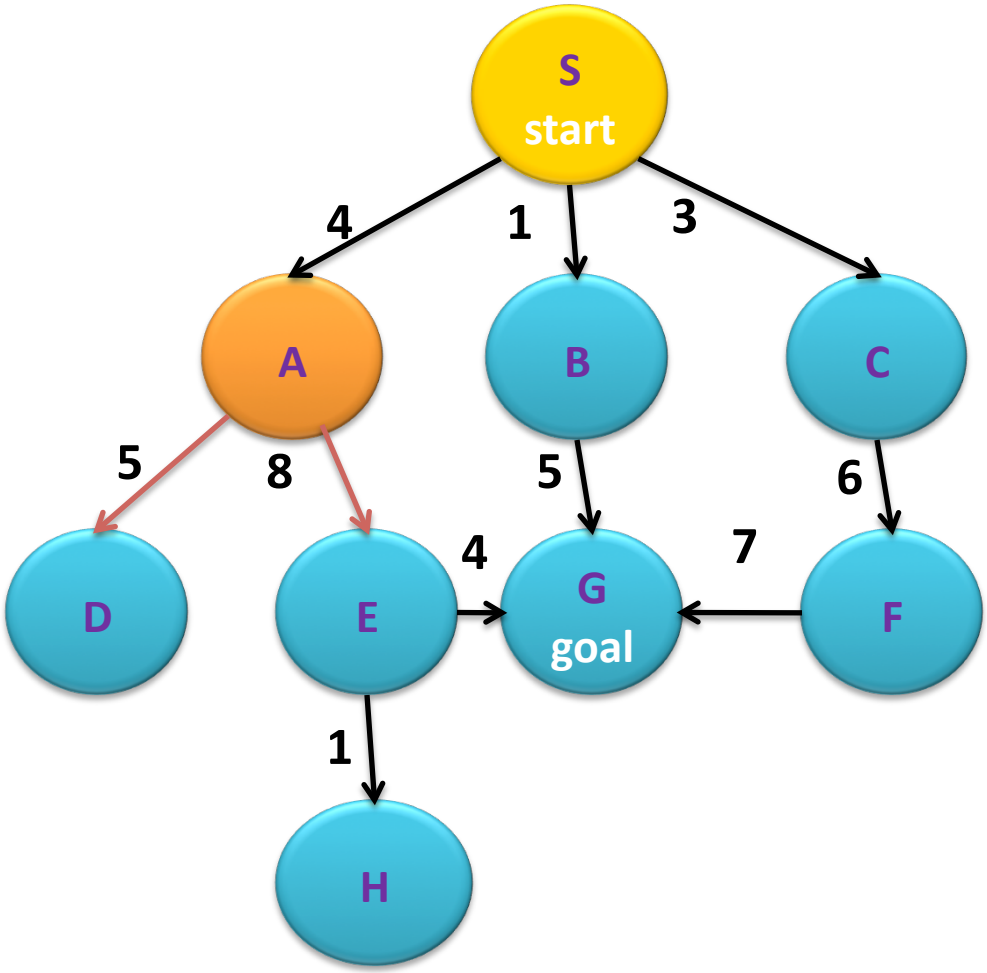


Depth-First Search(DFS)

General Search(Problem: stack)

#of nodes tested:2 expanded:2

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A not goal	{D, E B, C}

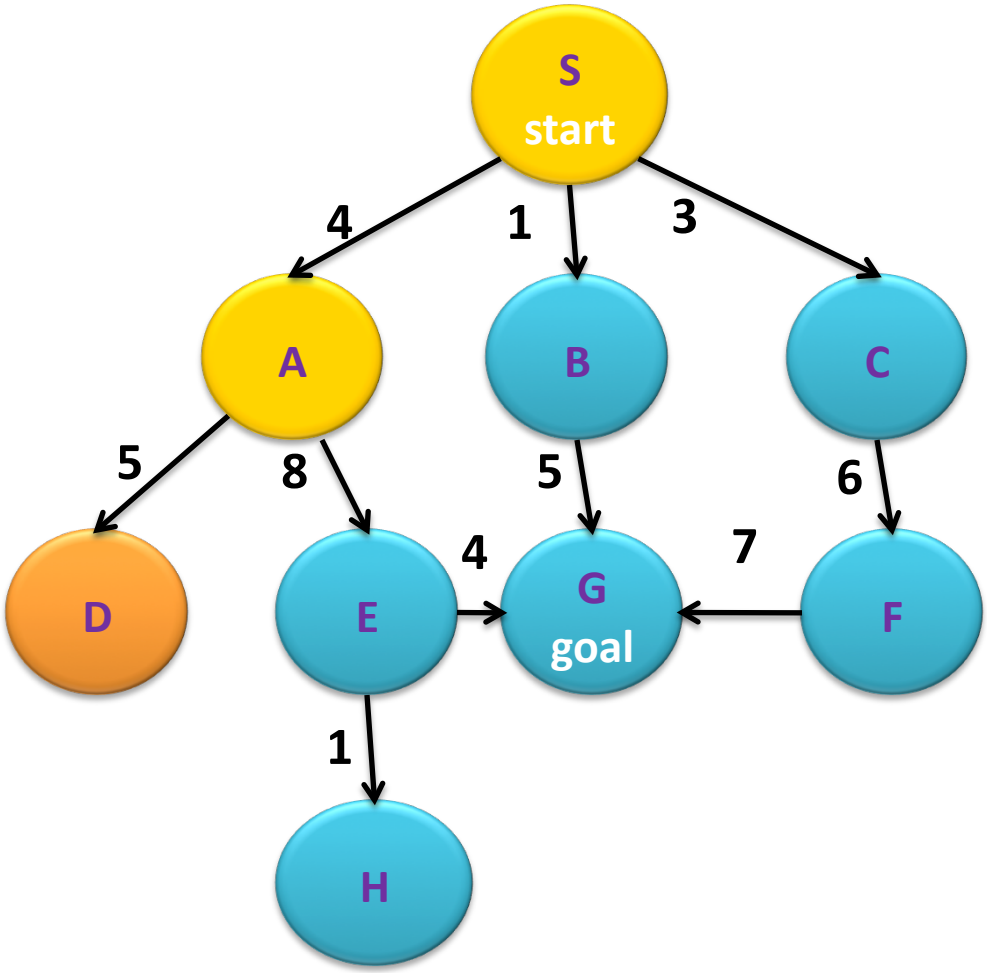


Depth-First Search(DFS)

General Search(Problem: stack)

#of nodes tested:3 expanded:2

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{D, E, B,C}
D not goal	{E, B,C} no expand

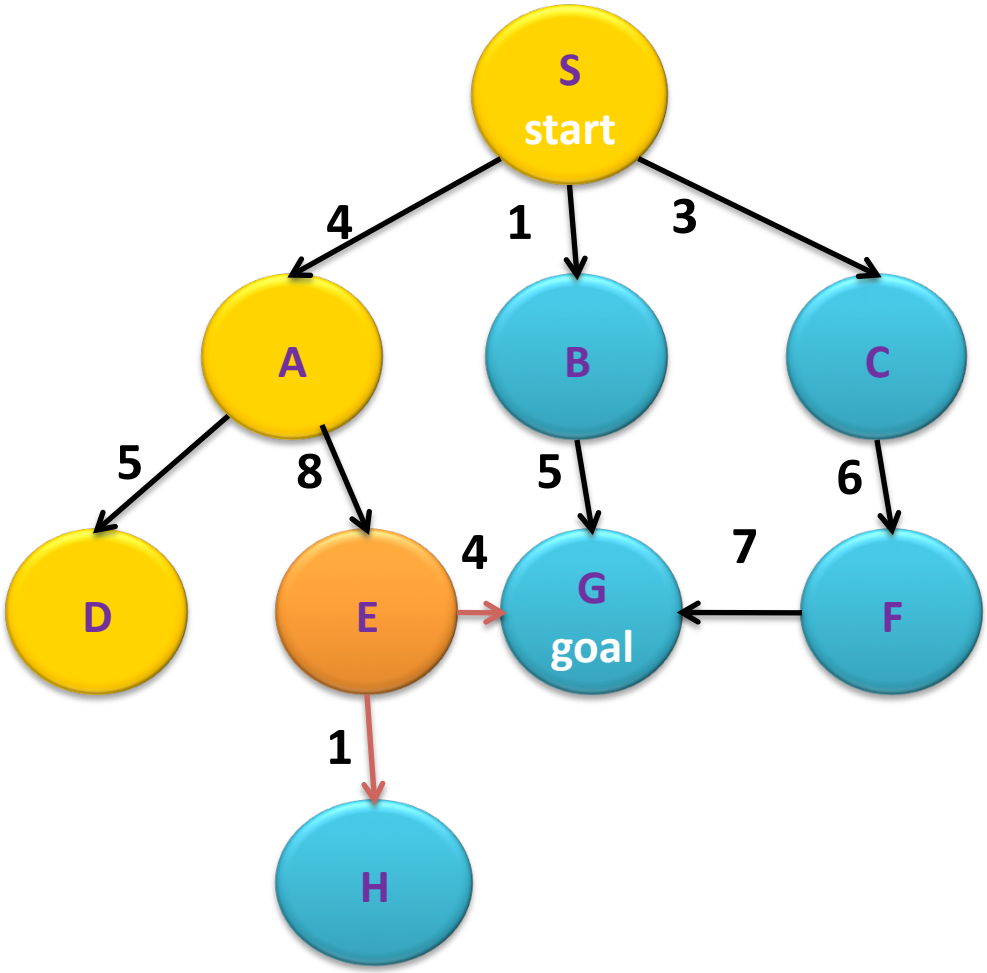


Depth-First Search(DFS)

General Search(Problem: stack)

#of nodes tested:4 expanded:3

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{D, E, B,C}
D	{E, B,C}
E not goal	{H,G,B, C}

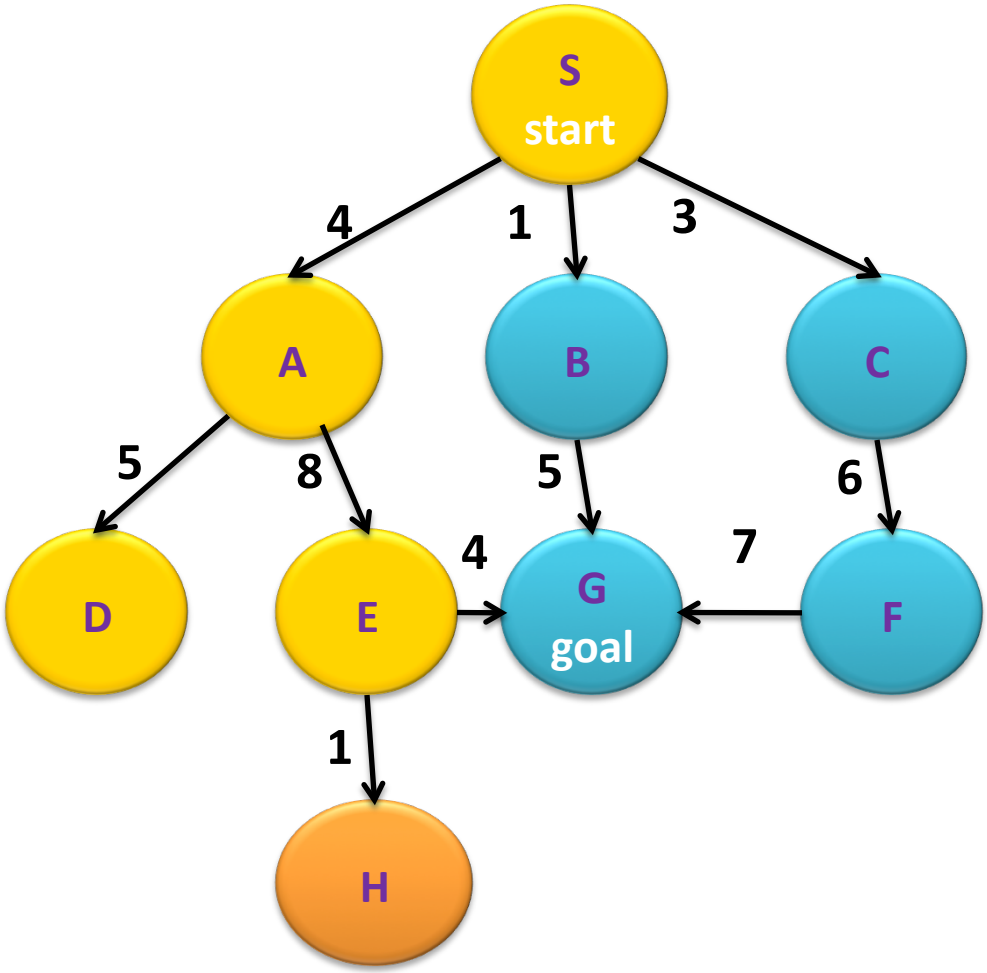


Depth-First Search(DFS)

General Search(Problem: stack)

#of nodes tested:5 expanded:3

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{D, B, C}
D	{B, C}
E	{H, G, G, C}
H not goal	{G, G, C} no expand

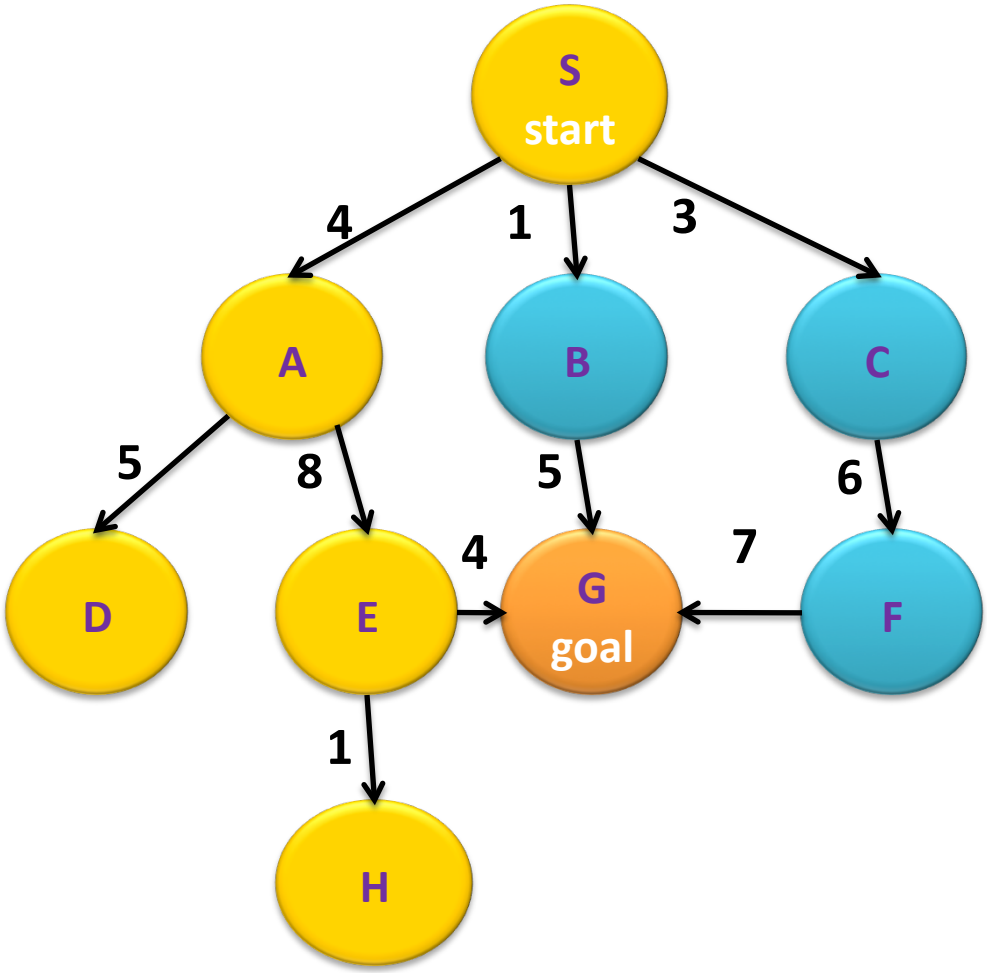


Depth-First Search(DFS)

General Search(Problem: stack)

#of nodes tested:6 expanded:3

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{D, B, C}
D	{B, C}
E	{H, G, G, C}
H	{G, G, C}
G goal	{G,C} no expand

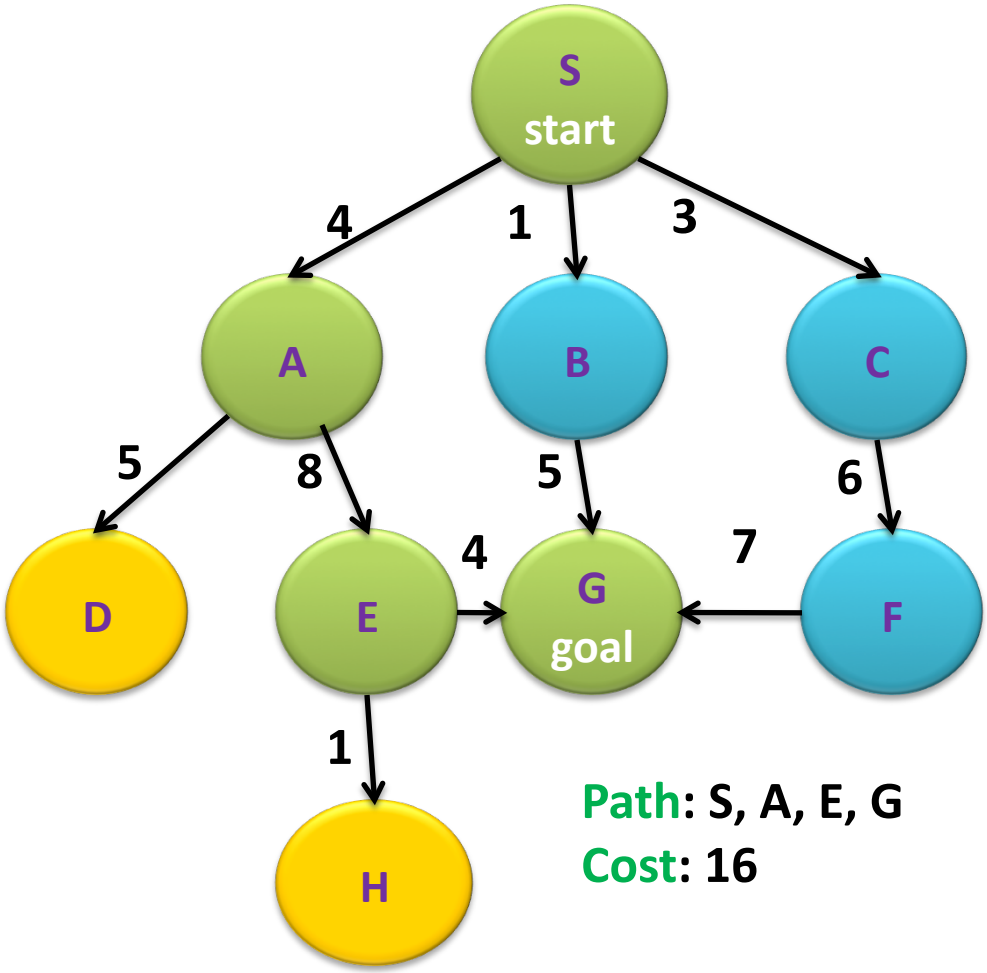


Depth-First Search(DFS)

General Search(Problem: stack)

#of nodes tested:6 expanded:3

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{D, B, C}
D	{B, C}
E	{H, G, G, C}
H	{G, G, C}
G	{G,C}



Depth-First Search(DFS)

- May not terminate without a depth bound
  i.e., cutting off search below a fixed depth, D

- Not complete
  – with or without cycle detection
  – and, with or without a depth cutoff

- Not optimal / admissible

*Can find long solutions quickly if lucky

Depth-First Search(DFS)

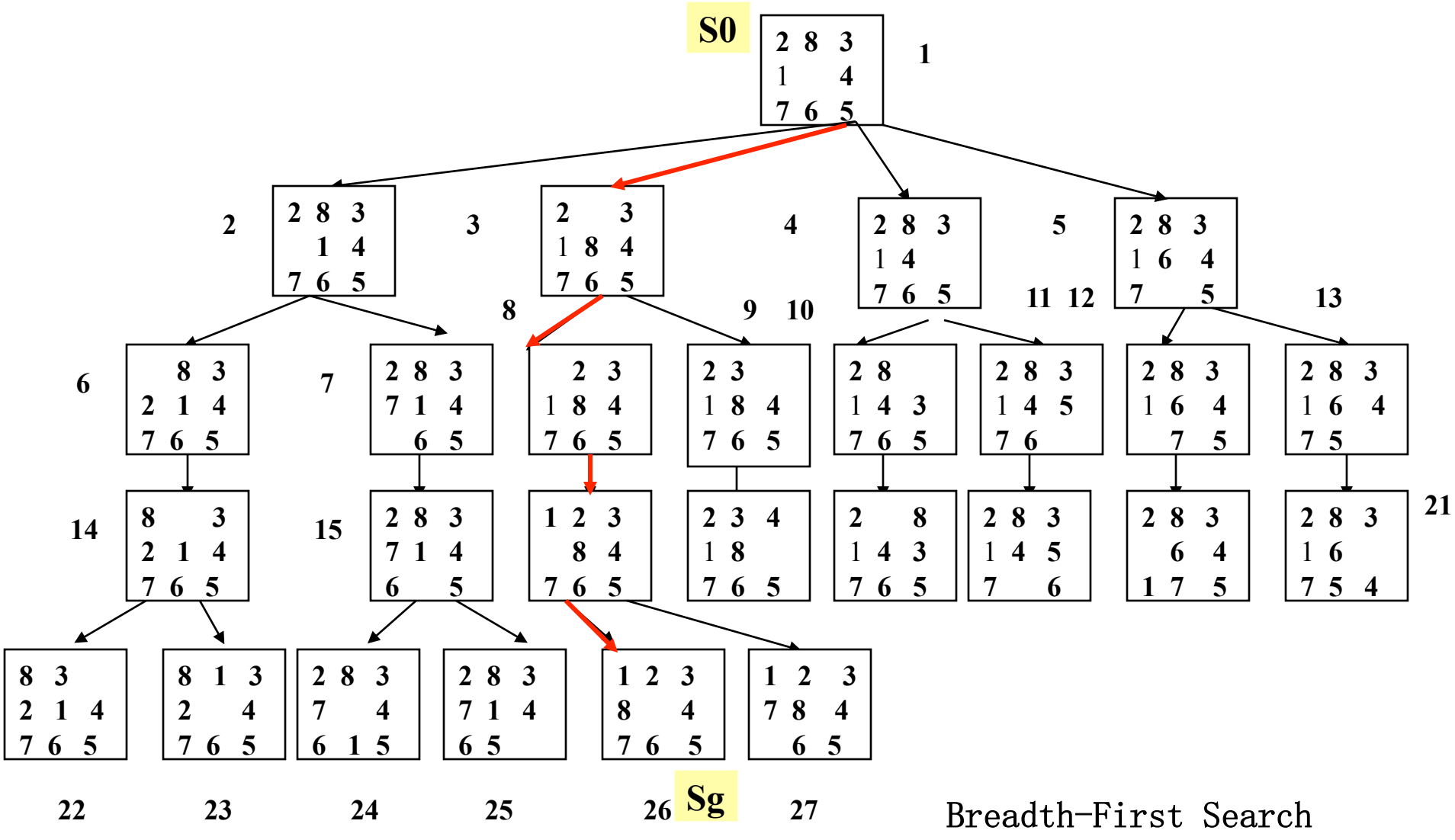
- Time complexity: $O(b^d)$ exponential

  Space complexity: $O(bd)$ linear

  – d is the depth of the solution

  – b is the branching factor at each non-leaf node

8 Digit Problem



Breadth-First Search

Uniform-Cost Search(UCS)

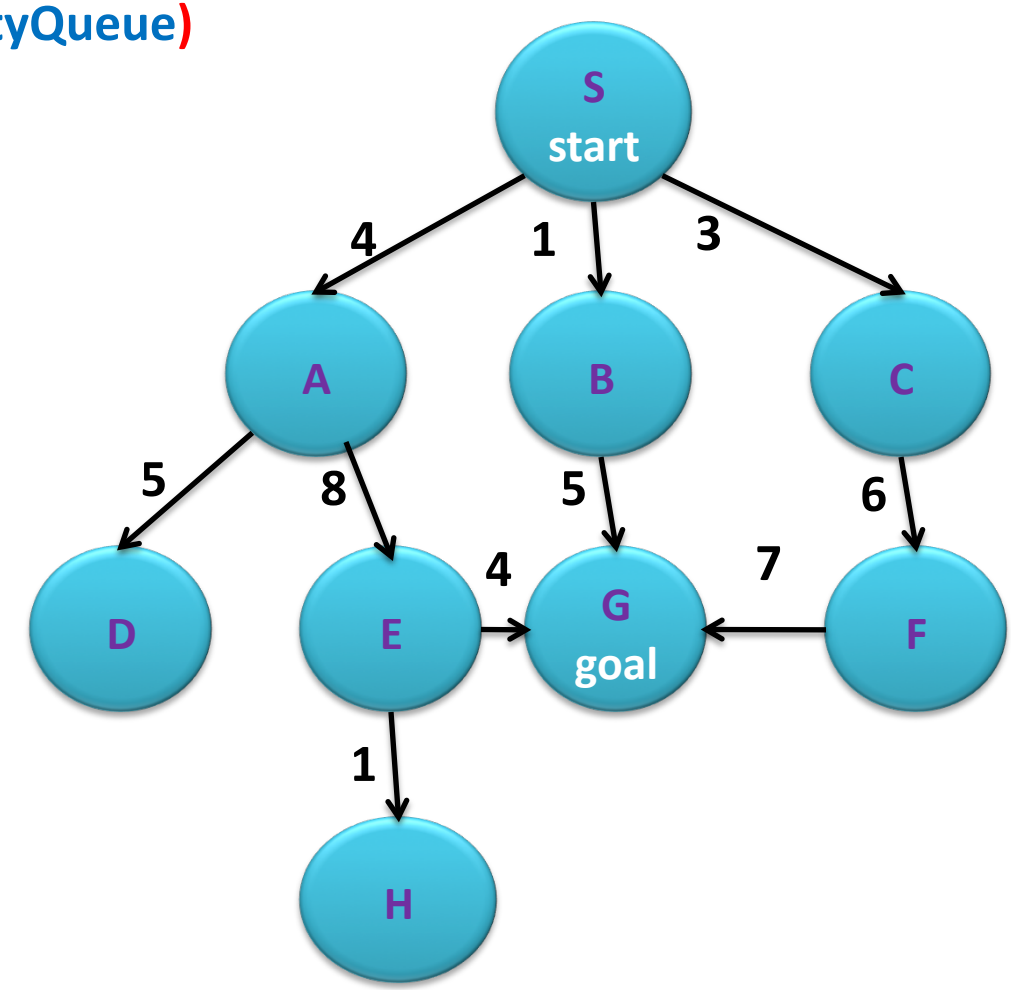
* priority queue used to order nodes, sort by path cost
  - let g(n) = cost of path from start node s to current node n
  - sort nodes by increasing value of g
  - only uninformed search that worries about costs

Uniform-Cost Search(UCS)

General Search(Problem: priorityQueue)

#of nodes tested:0 expanded:0

Expnd. node	Fringe list
	{S}

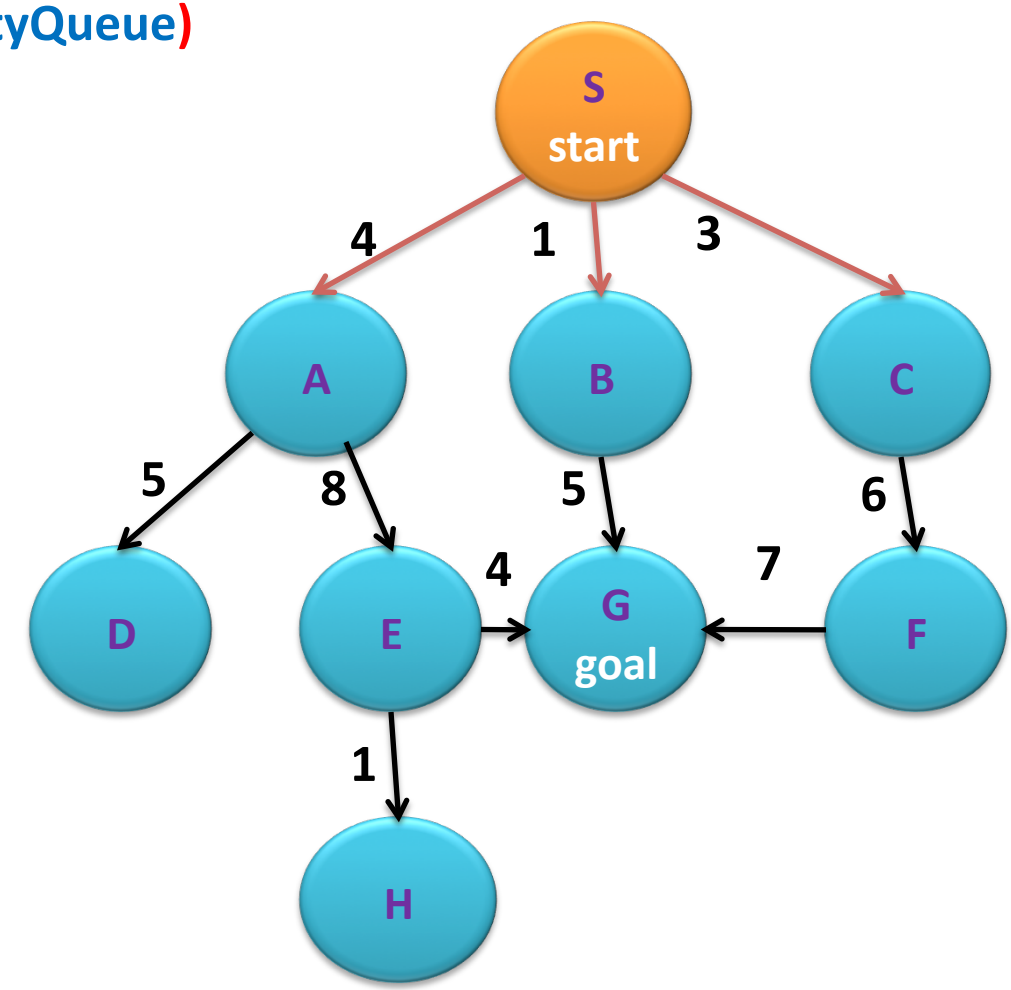


Uniform-Cost Search(UCS)

General Search(Problem: priorityQueue)

#of nodes tested:1 expanded:1

Expnd. node	Fringe list
	{S}
S not goal	{B: 1,C: 3, A: 4}

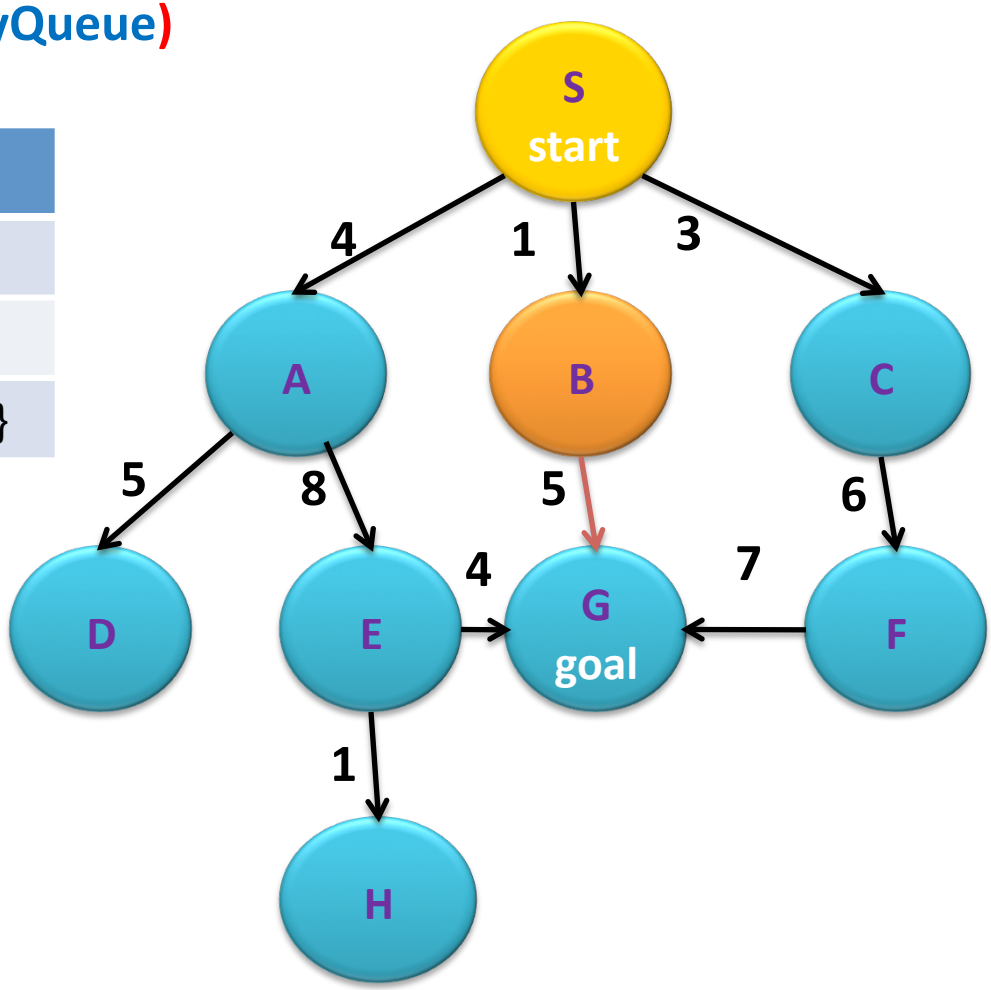


Uniform-Cost Search(UCS)

General Search(Problem: priorityQueue)

#of nodes tested:2 expanded:2

Expnd. node	Fringe list
	{S}
S	{B: 1,C: 3, A: 4}
B not goal	{C: 3, A: 4, G: 1+5 }

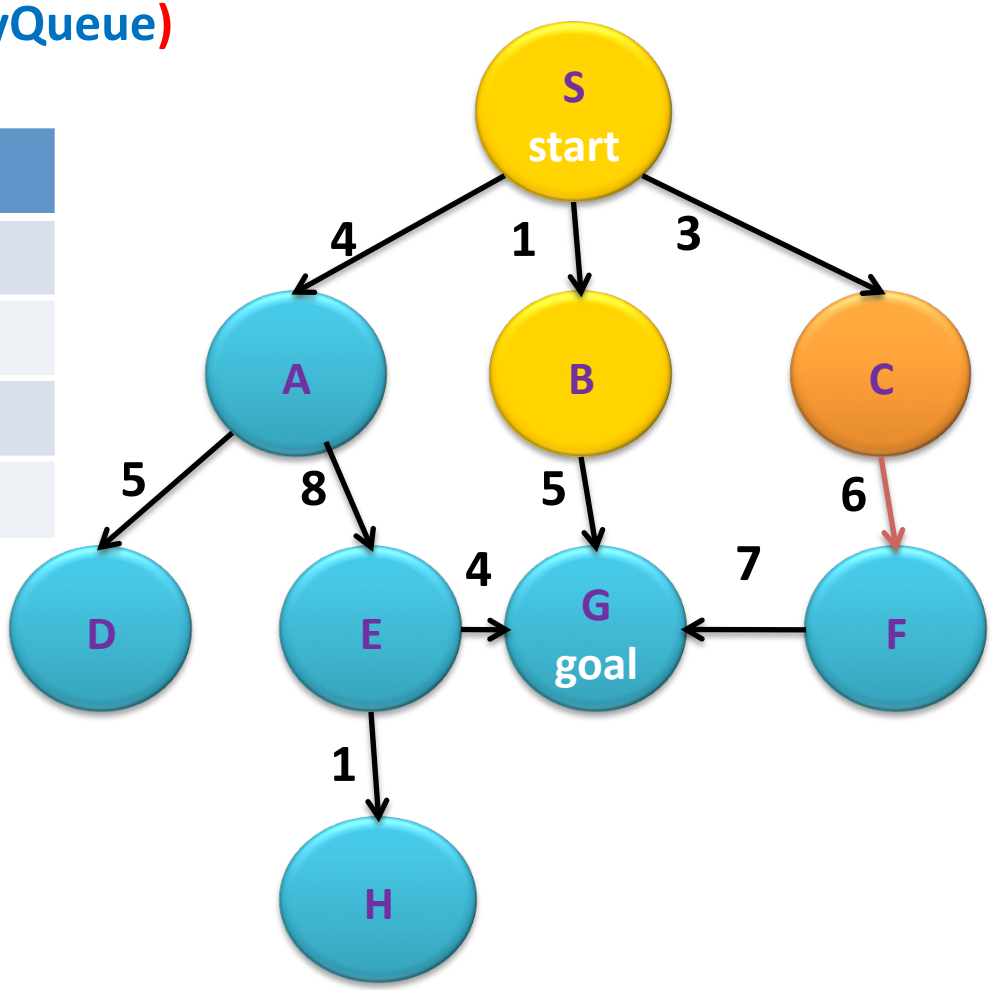


Uniform-Cost Search(UCS)

General Search(Problem: priorityQueue)

#of nodes tested:3 expanded:3

Expnd. node	Fringe list
	{S}
S	{B:1,C:3, A:4}
B	{C:3, A:4, G:1+5 }
C not goal	{A:4, G:6, F:9 }

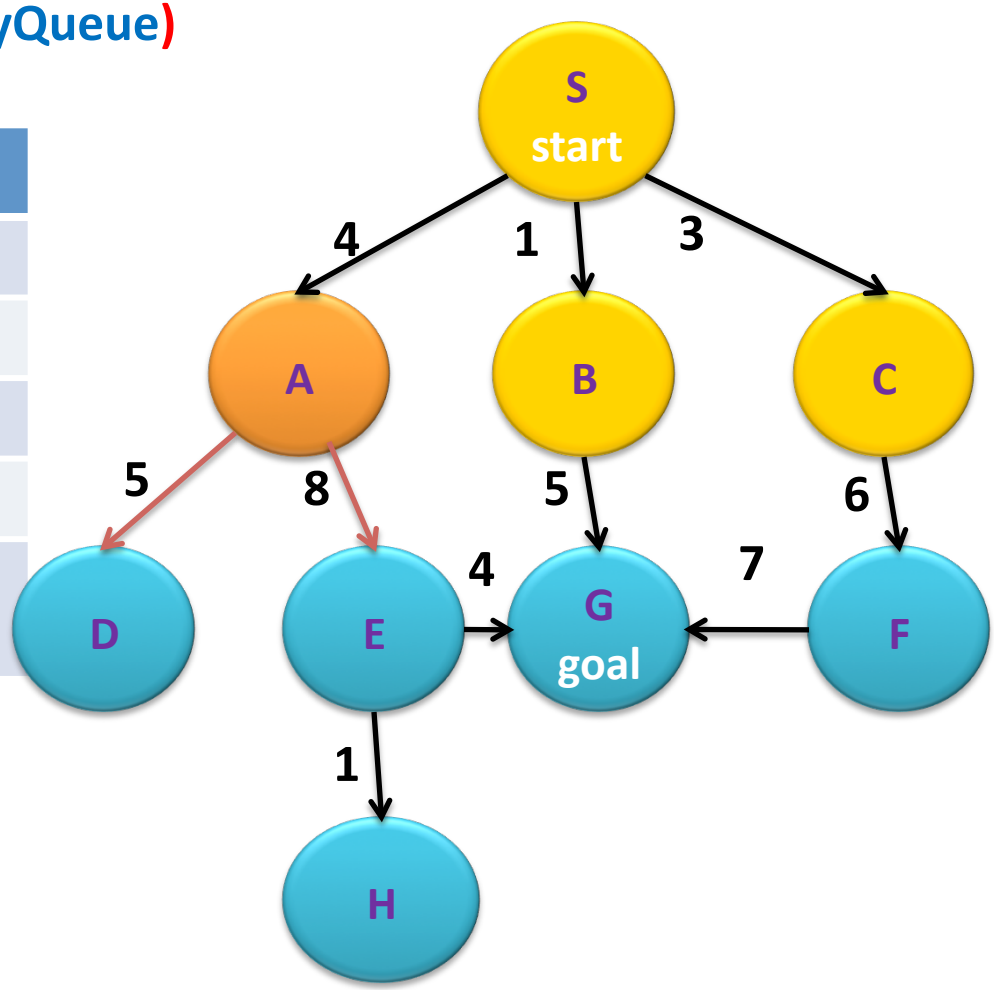


Uniform-Cost Search(UCS)

General Search(Problem: priorityQueue)

#of nodes tested:4 expanded:4

Expnd. node	Fringe list
	{S}
S	{B:1,C:3, A:4}
B	{C:3, A:4, G:1+5 }
C	{A:4, G:6, F:9 }
A not goal	{G:6, F:9, D:9, E: 12}

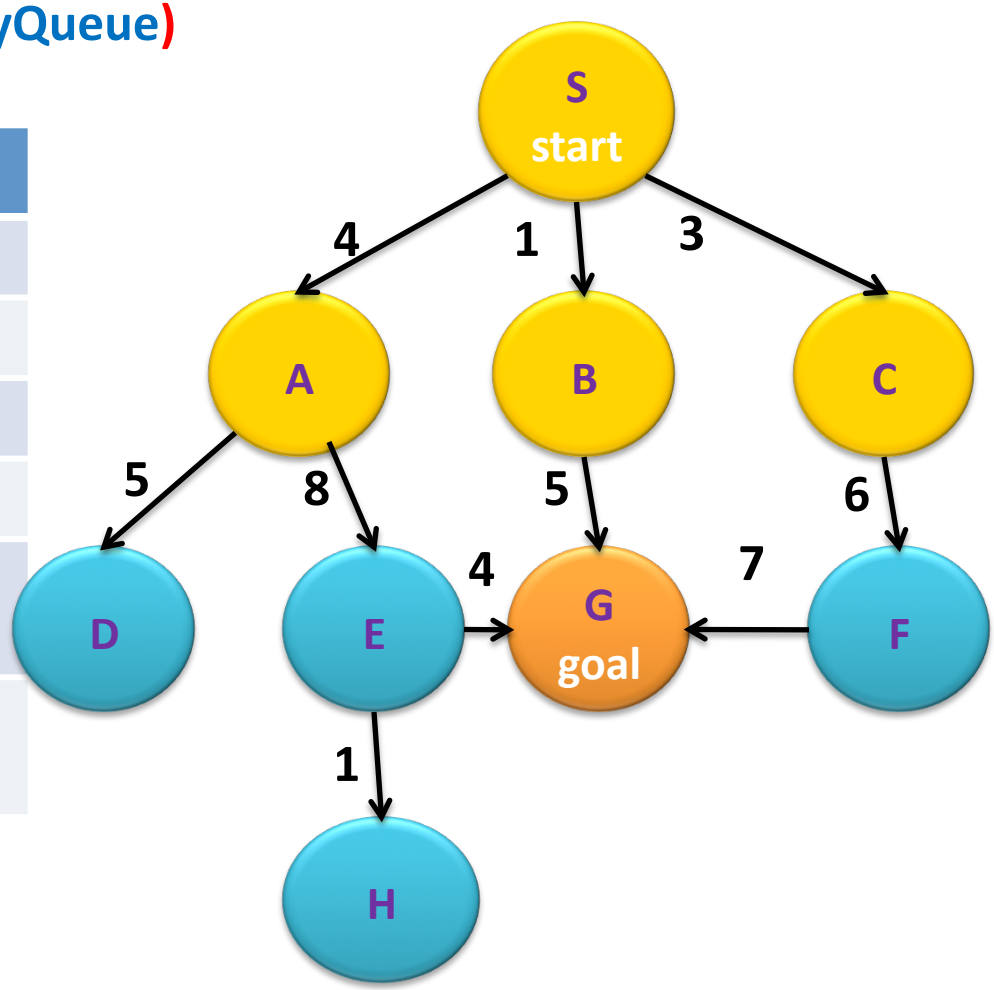


Uniform-Cost Search(UCS)

General Search(Problem: priorityQueue)

#of nodes tested:5 expanded:4

Expnd. node	Fringe list
	{S}
S	{B:1,C:3, A:4}
B	{C:3, A:4, G:1+5 }
C	{A:4, G:6, F:9 }
A	{G:6, F:9, D:9, E: 12}
G goal	{F:9, D:9, E:12} no expand

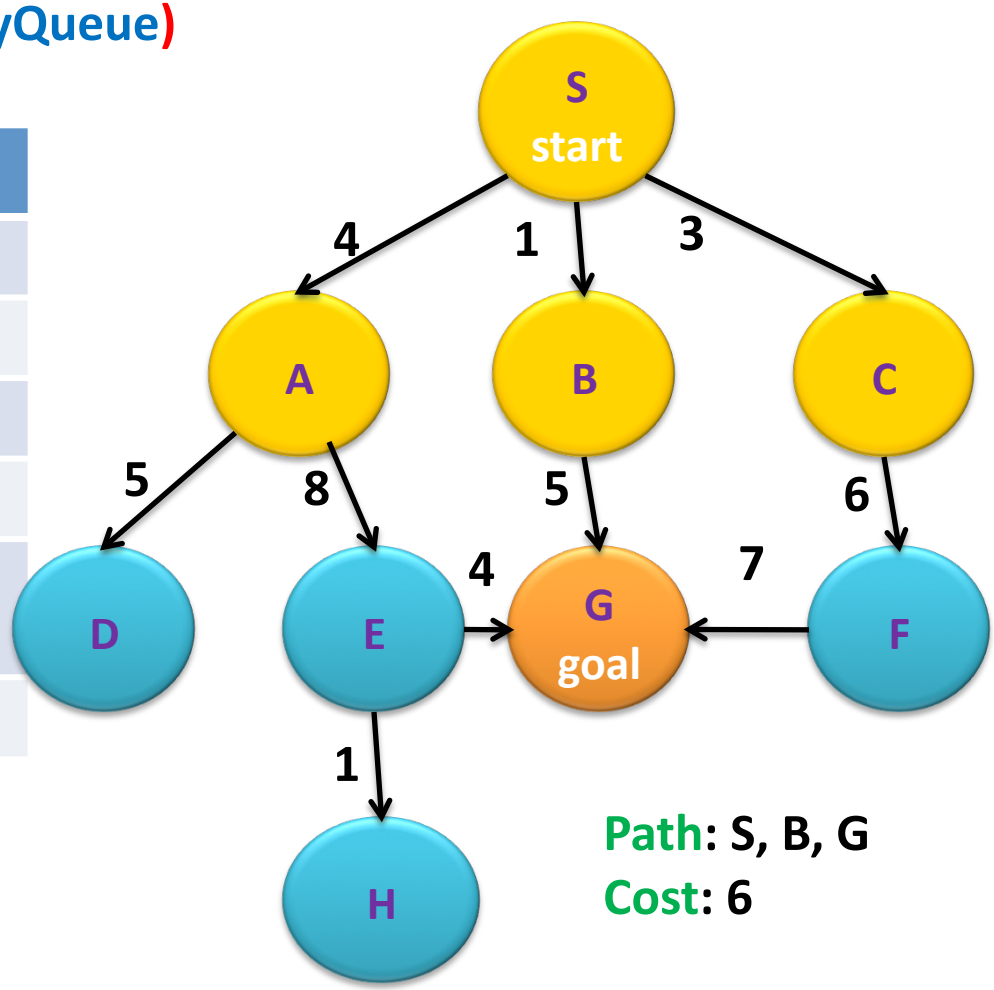


Uniform-Cost Search(UCS)

General Search(Problem: priorityQueue)

#of nodes tested:5 expanded:4

Expnd. node	Fringe list
	{S}
S	{B:1,C:3, A:4}
B	{C:3, A:4, G:1+5 }
C	{A:4, G:6, F:9 }
A	{G:6, F:9, D:9, E: 12}
G	{F:9, D:9, E:12}



Uniform-Cost Search(UCS)

- Called Dijkstra's Algorithm in the algorithms lit.
- Complete
- Optimal / Admissible

  – requires that the goal test being applied when a node is ***removed*** from the Fringe list rather than when the node is first generated while its parent node is expanded
- Time and space complexity: $O(b^d)$ exponential

Depth-First, Iterative-Deepening Search(IDS)

requires modification to search algorithm:

- do DFS to depth 1

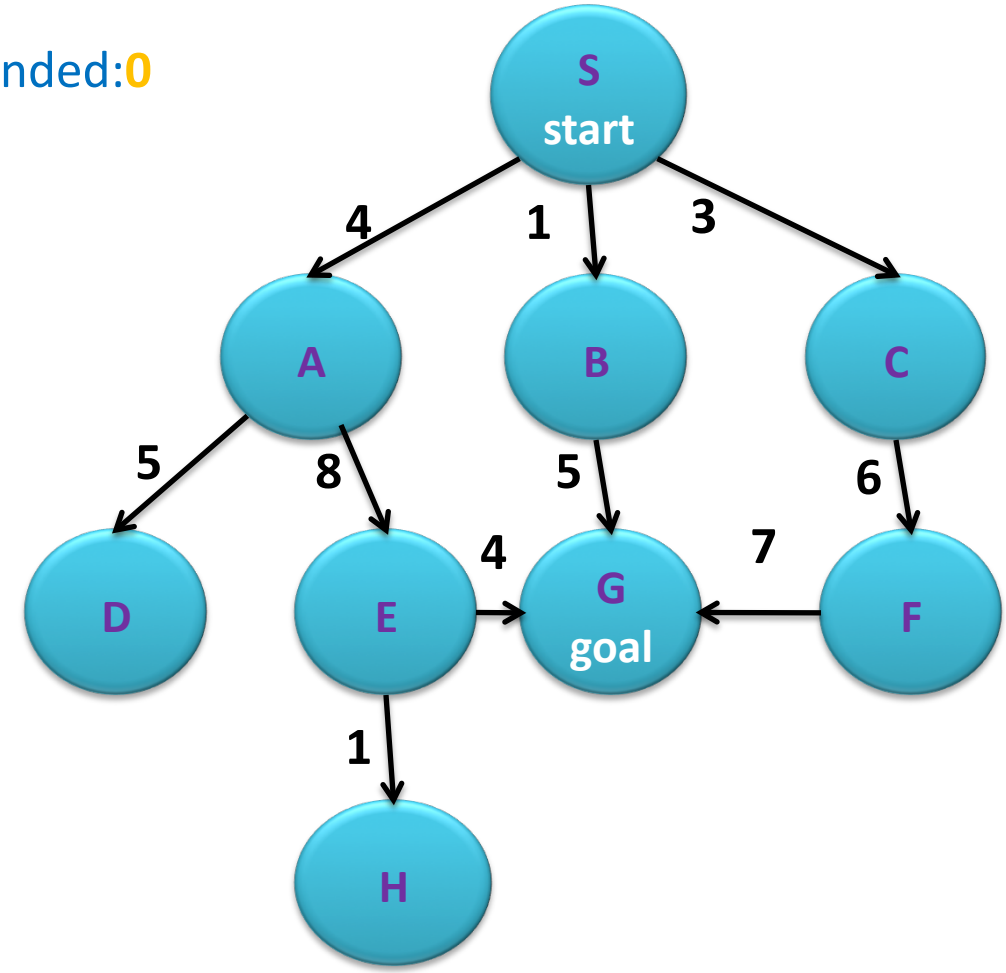
  treat all children of the start node as leafs
- if no solution found, do DFS to depth 2
- repeat by increasing depth until a solution found

Depth-First, Iterative-Deepening Search(IDS)

General Search

Depth: 1#of nodes tested:0 expanded:0

Expnd. node	Fringe list
	{S}

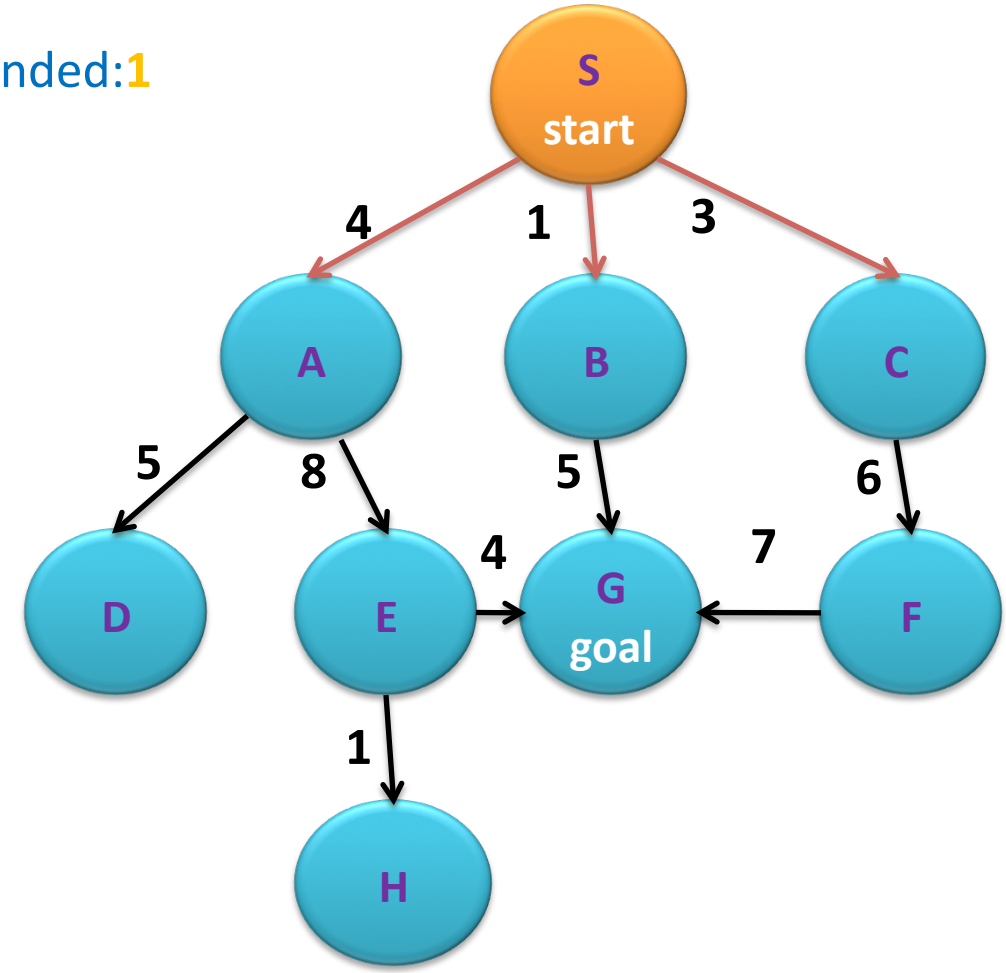


Depth-First, Iterative-Deepening Search (IDS)

General Search

Depth: 1#of nodes tested:1 expanded:1

Expnd. node	Fringe list
	{S}
S not goal	{A,B,C}

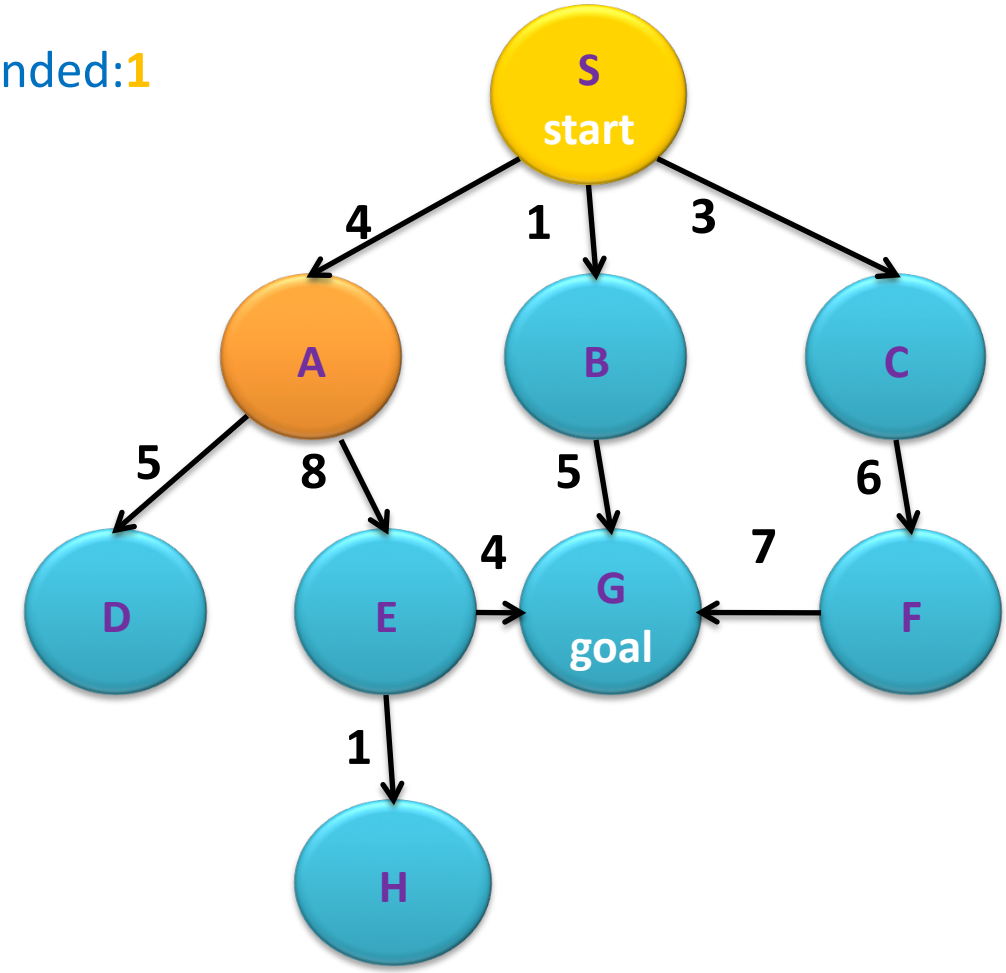


Depth-First, Iterative-Deepening Search (IDS)

General Search

Depth: 1#of nodes tested:2 expanded:1

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A not goal	{B,C} no expand

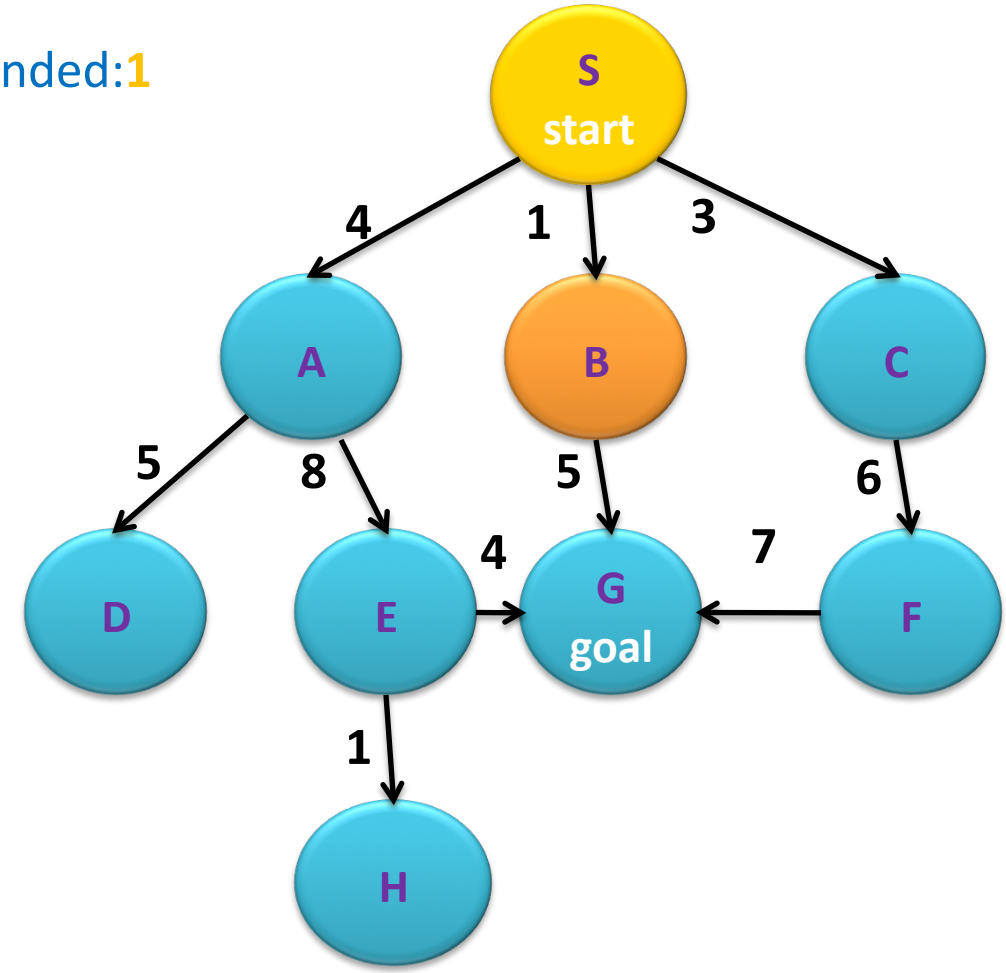


Depth-First, Iterative-Deepening Search (IDS)

General Search

Depth: 1#of nodes tested:3 expanded:1

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{B,C}
B not goal	{C} no expand

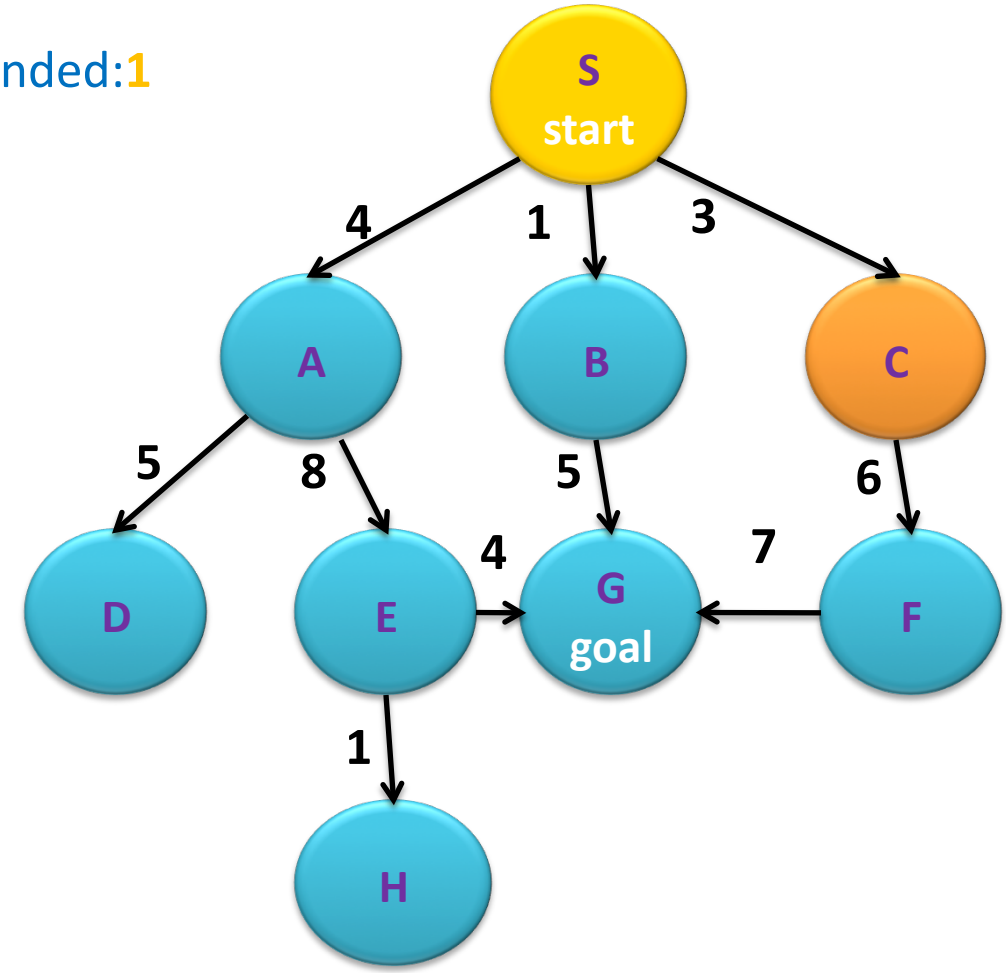


Depth-First, Iterative-Deepening Search (IDS)

General Search

Depth: 1#of nodes tested:4 expanded:1

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{B,C}
B	{C}
C not goal	{} no expand FAIL

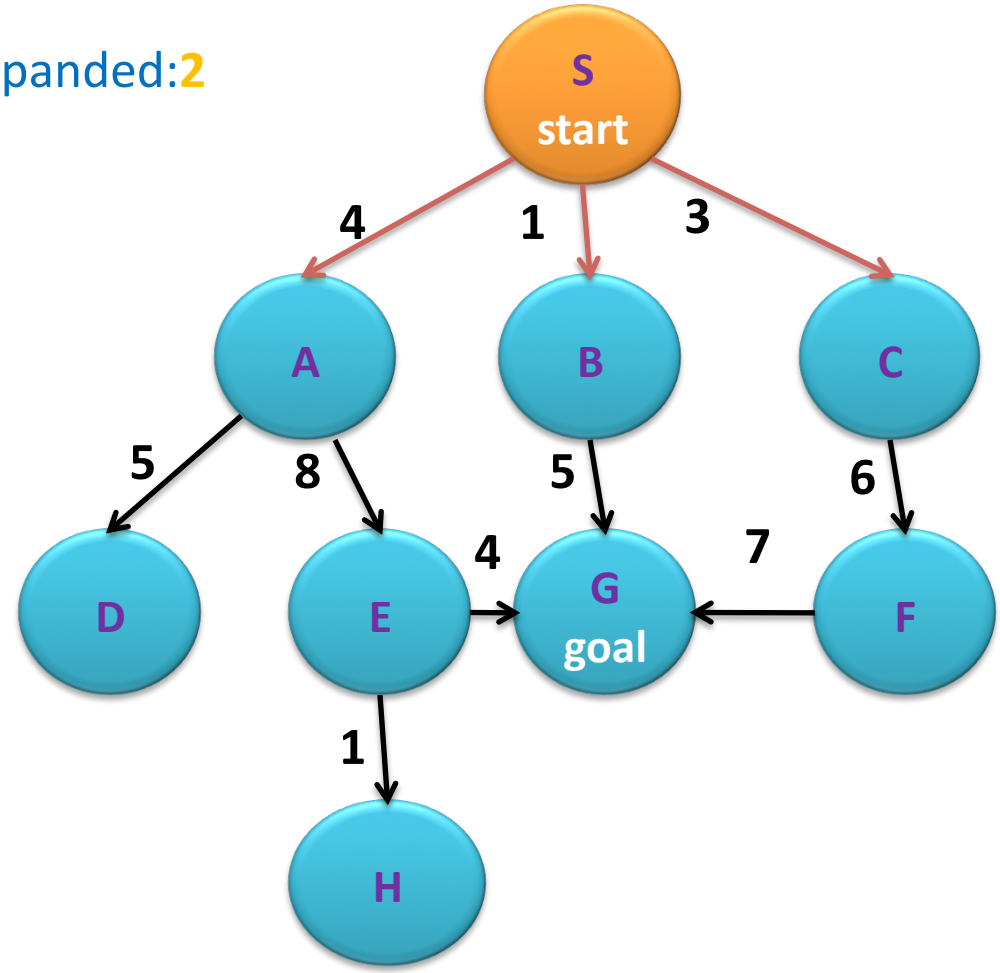


Depth-First, Iterative-Deepening Search (IDS)

General Search

Depth: 2#of nodes tested:4(1) expanded:2

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{B,C}
B	{C}
C	{}
S no test	{A,B,C}

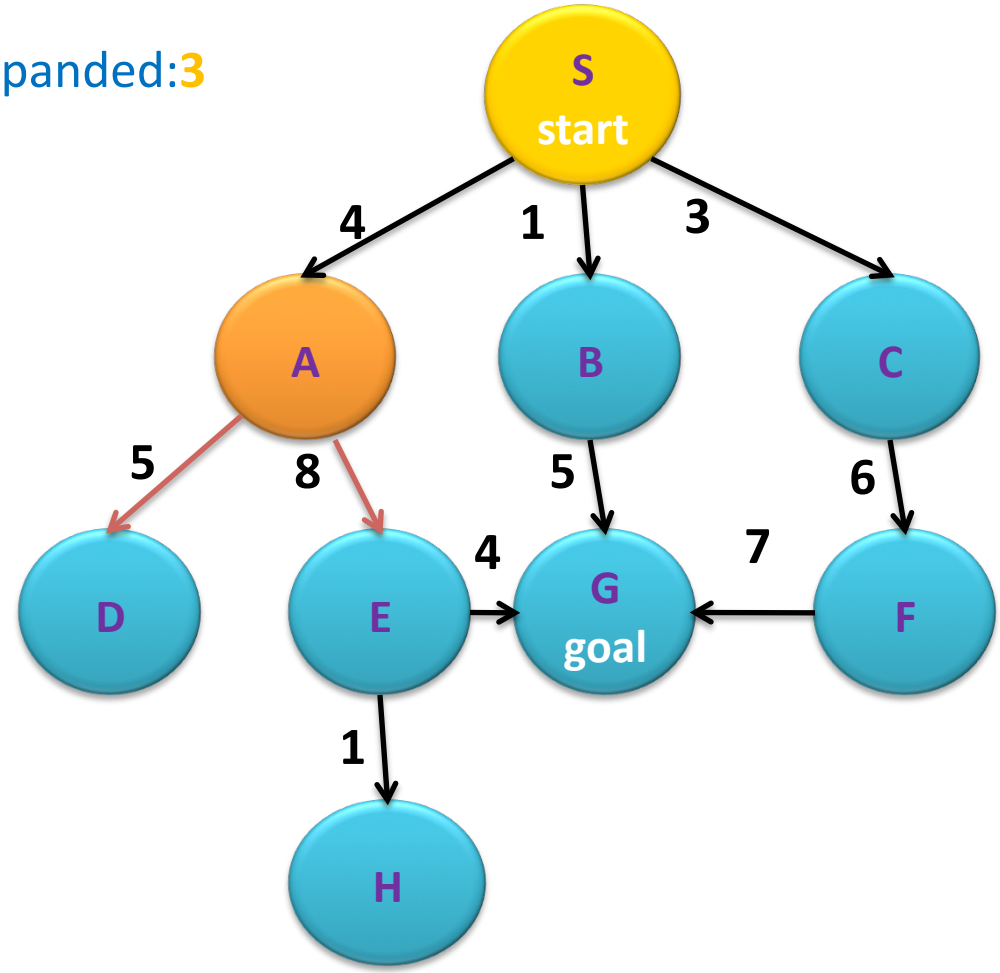


Depth-First, Iterative-Deepening Search (IDS)

General Search

Depth: 2#of nodes tested:4(2) expanded:3

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{B,C}
B	{C}
C	{}
S	{A,B,C}
A no test	{D,E,B,C}

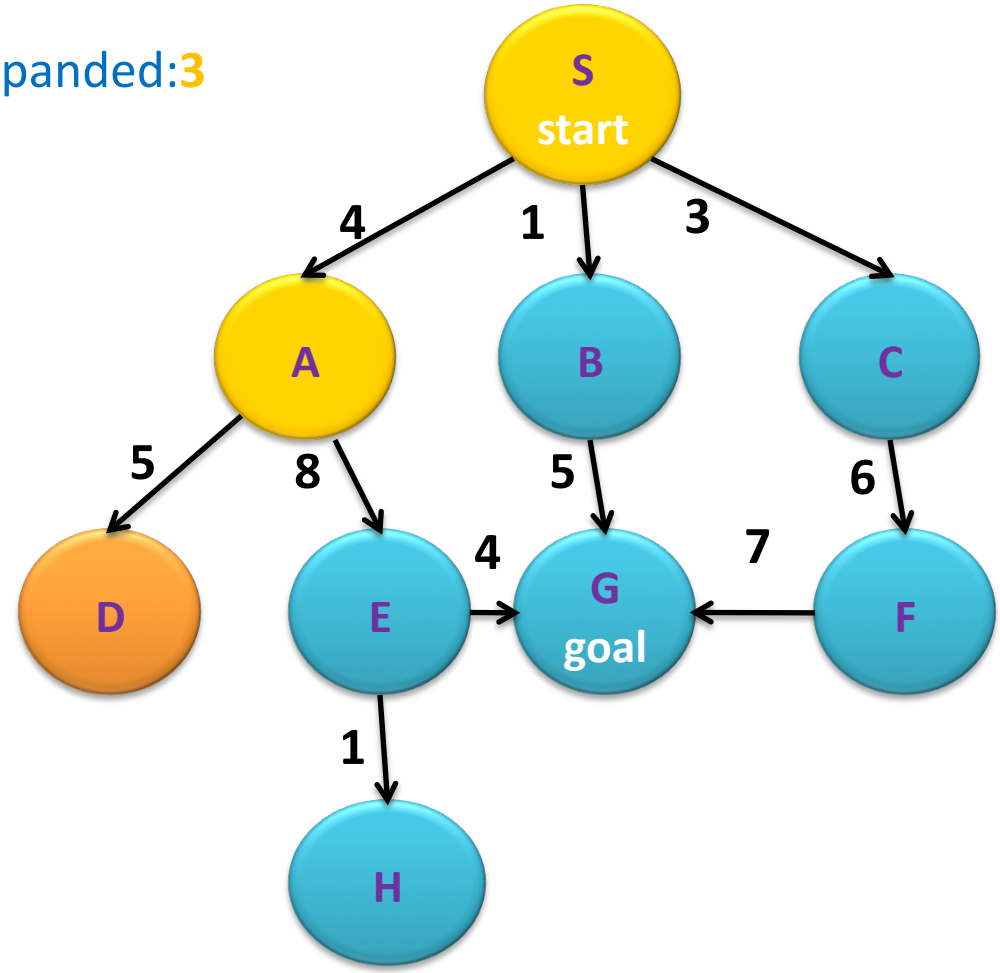


Depth-First, Iterative-Deepening Search (IDS)

General Search

Depth: 2#of nodes tested:5(2) expanded:3

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{B,C}
B	{C}
C	{}
S	{A,B,C}
A	{D,E,B,C}
D not goal	{E,B,C} no expand

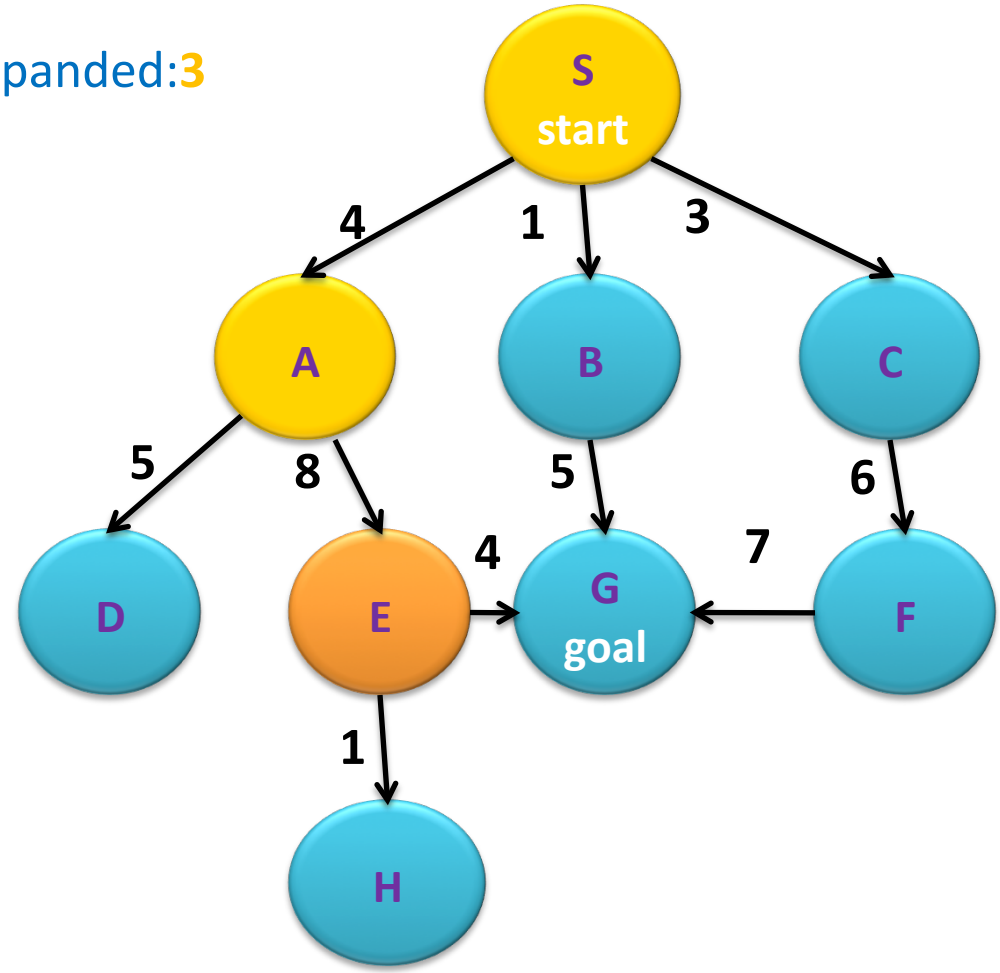


Depth-First, Iterative-Deepening Search (IDS)

General Search

Depth: 2#of nodes tested:6(2) expanded:3

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{B,C}
B	{C}
C	{}
S	{A,B,C}
A	{D,E,B,C}
D	{E,B,C}
E not goal	{B,C} no expand

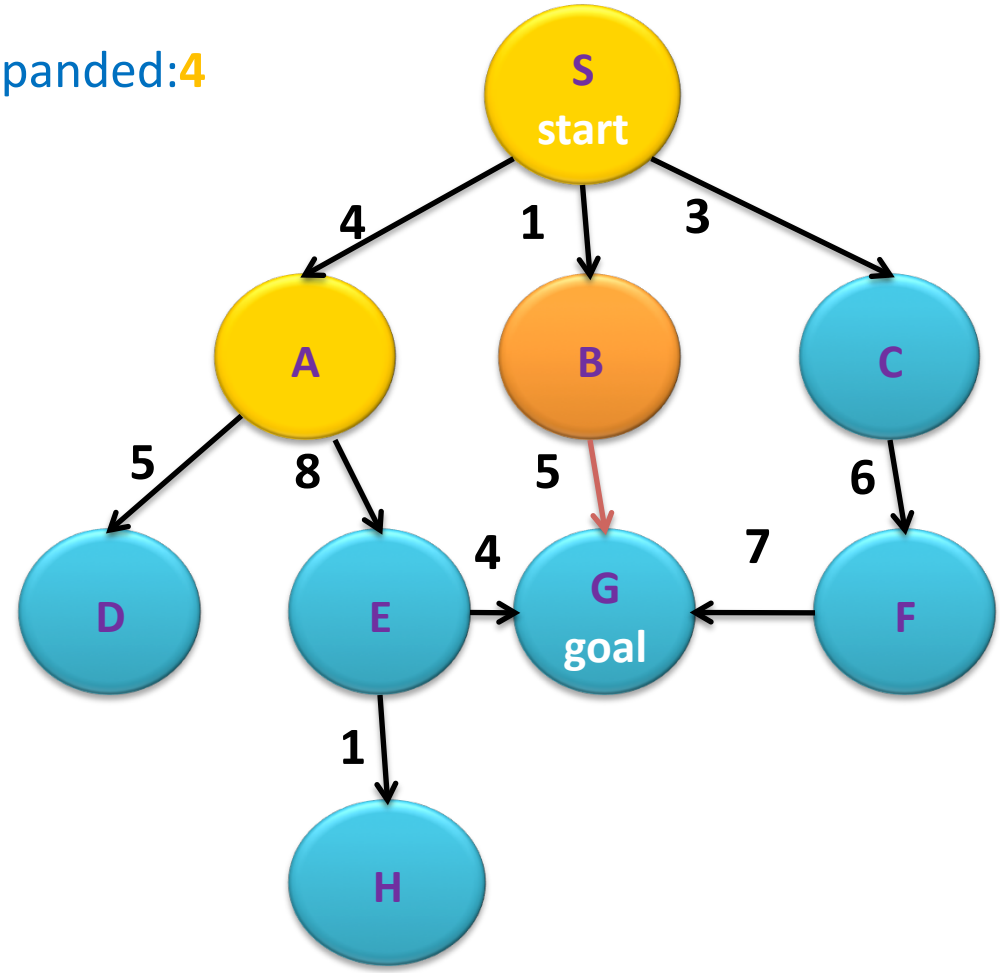


Depth-First, Iterative-Deepening Search (IDS)

General Search

Depth: 2#of nodes tested:6(3) expanded:4

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{B,C}
B	{C}
C	{}
S	{A,B,C}
A	{D,E,B,C}
D	{E,B,C}
E	{B,C}
B no test	{G,C}

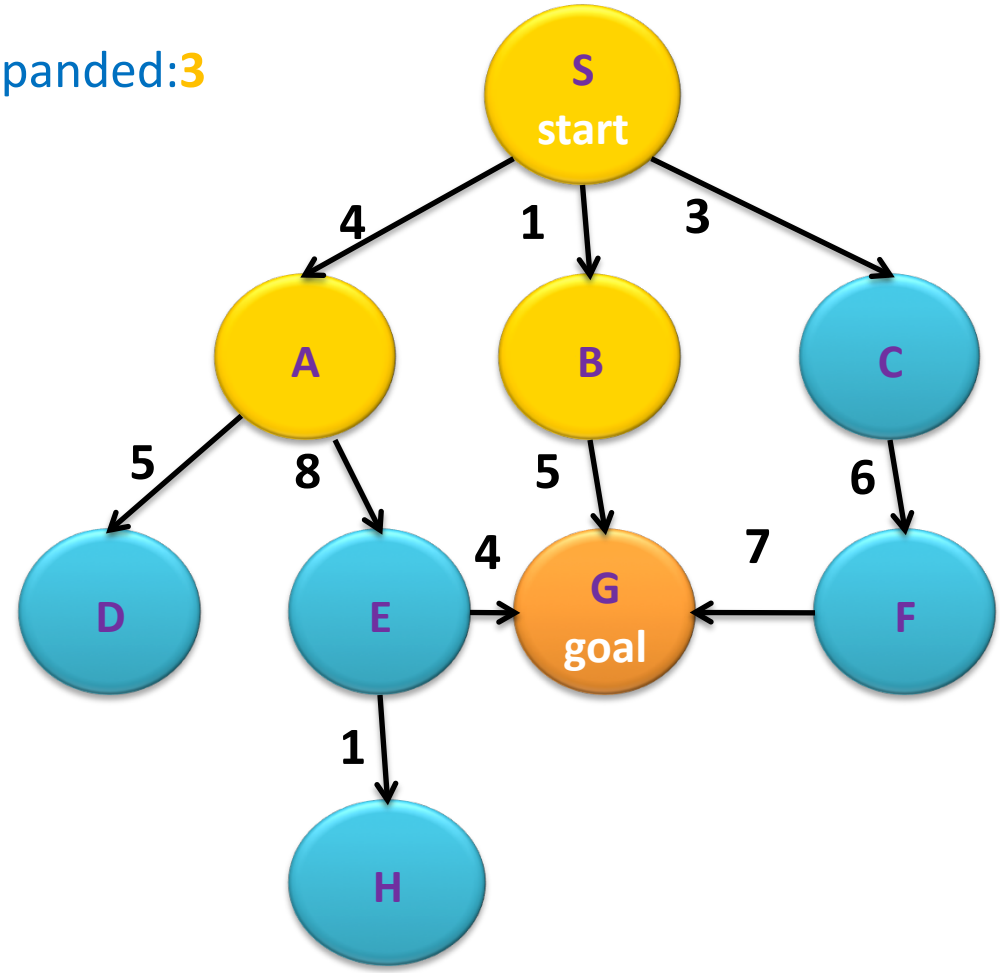


Depth-First, Iterative-Deepening Search (IDS)

General Search

Depth: 2#of nodes tested:7(3) expanded:3

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{B,C}
B	{C}
C	{}
S	{A,B,C}
A	{D,E,B,C}
D	{E,B,C}
E	{B,C}
B	{G,C}
G goal	{C} no expand

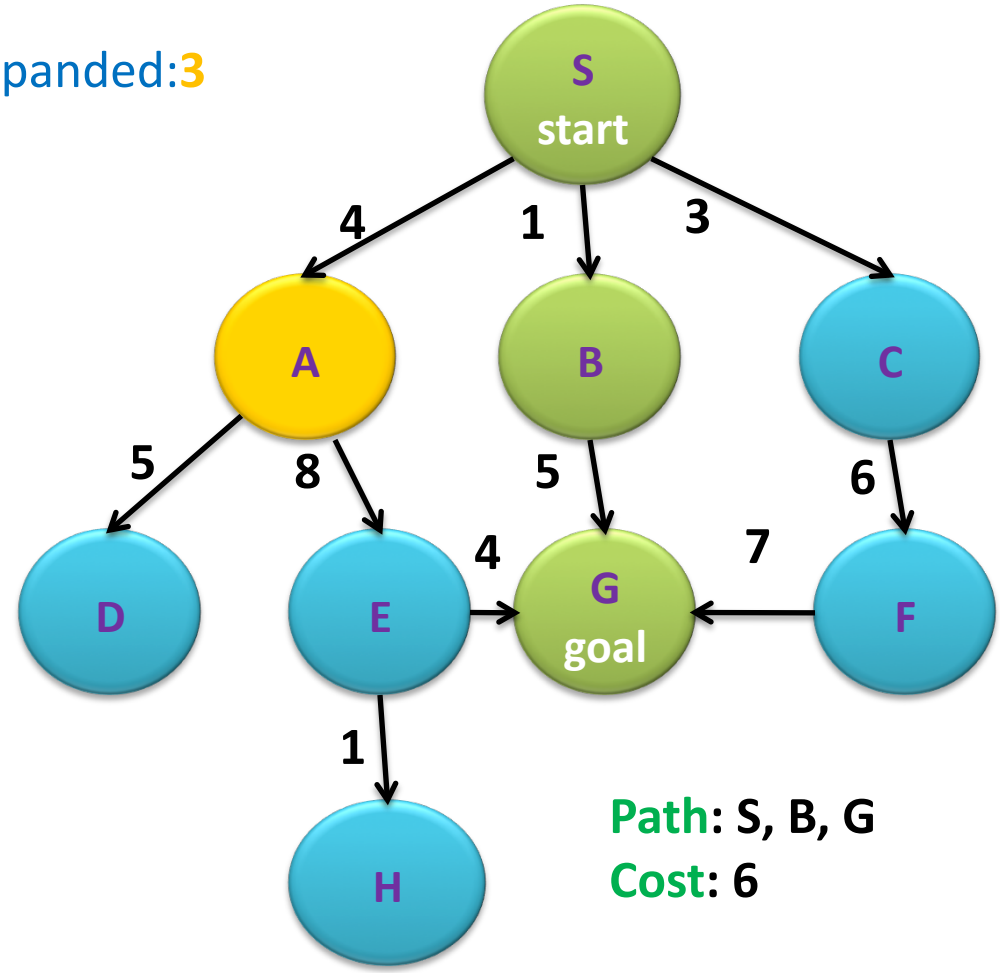


Depth-First, Iterative-Deepening Search (IDS)

General Search

Depth: 2#of nodes tested:7(3) expanded:3

Expnd. node	Fringe list
	{S}
S	{A,B,C}
A	{B,C}
B	{C}
C	{}
S	{A,B,C}
A	{D,E,B,C}
D	{E,B,C}
E	{B,C}
B	{G,C}
G	{C}



Depth-First, Iterative-Deepening Search (IDS)

- Has advantages of BFS
  - – completeness
  - – optimality as stated for BFS

- Has advantages of DFS
  - – limited space
  - – in practice, even with redundant effort in practice, even with redundant effort, ***it still finds longer paths more quickly than BFS***

Depth-First, Iterative-Deepening Search (IDS)

- Space complexity: $O(bd)$ (i.e., linear like DFS)

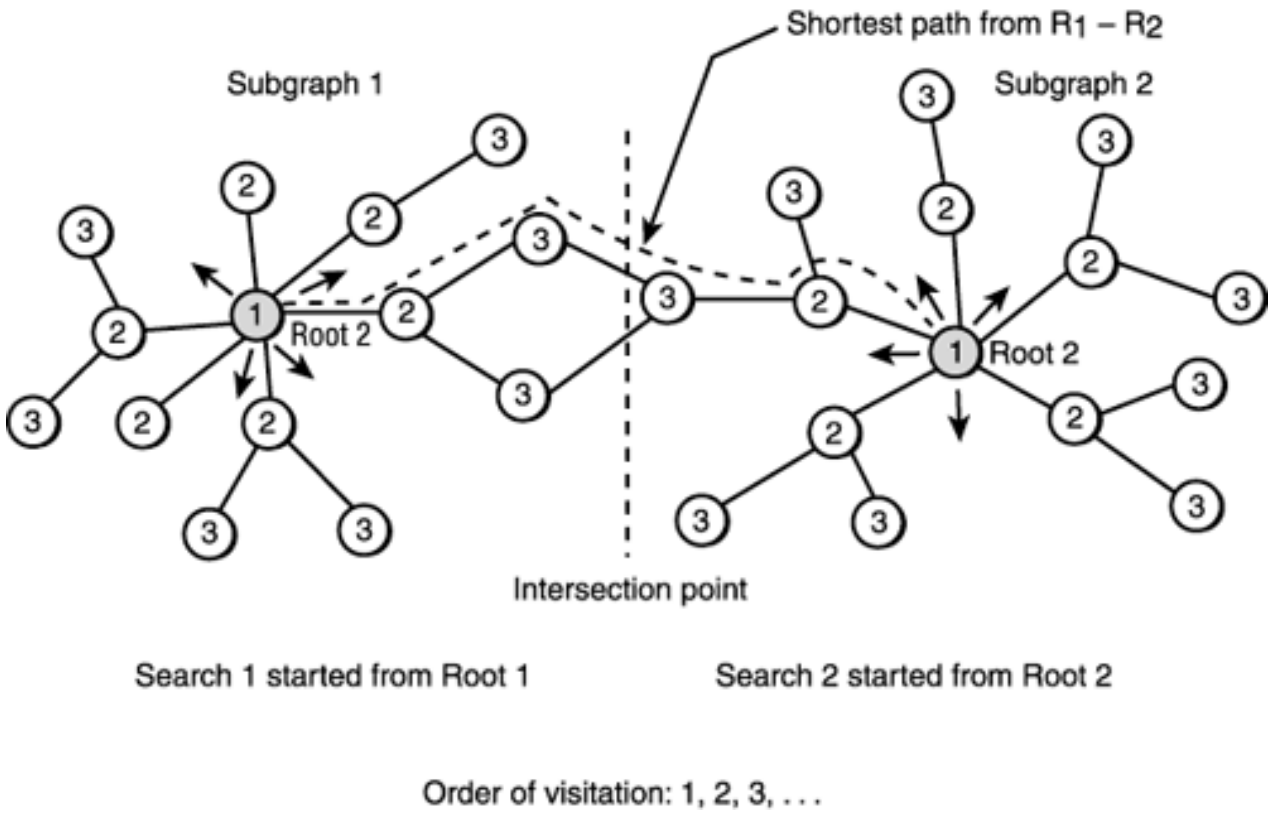
- Time complexity

  is a little worse than BFS or DFS

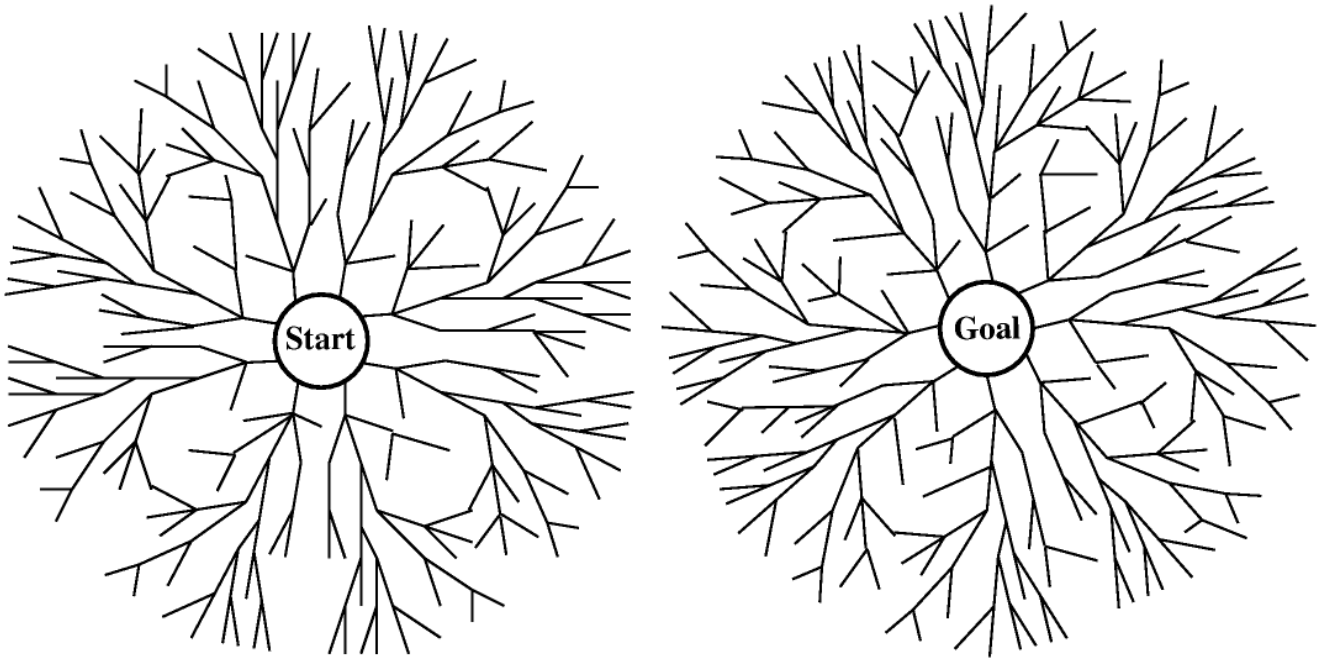
  because nodes near the top of the search tree are generated multiple times (redundant effort)

Bidirectional Search

- Breadth-first search from both start and goal
- Fringe meet
- Generates $O(b^{\frac{d}{2}})$ instead of $O(b^d)$ nodes



Bidirectional Search—Which direction should we search?



Our choices: Forward, backward, bidirectional?

Issues: How many start and goal states are there?

Branching factors in each direction

How much work is it to compare states?

Comparing uninformed search strategies

b: branching factor (assume finite) d: goal depth m: graph depth

	Complete	optimal	time	space
Breadth-first search	Y	Y, if [1]	$O(b^d)$	$O(b^d)$
Uniform-cost search[2]	Y	Y	$O(b^{C^*/\varepsilon})$	$O(b^{C^*/\varepsilon})$
Depth-first search	N	N	$O(b^m)$	O(bm)
Iterative deepening	Y	Y, if [1]	$O(b^d)$	O(bd)
Bidirectional search[3]	Y	Y, if [1]	$O(b^{d/2})$	$O(b^{d/2})$

1. Edge cost constant, or positive non-decreasing in depth
3. Edge costs $\geq \varepsilon > 0$. C^* is the best goal path cost
4. Both directions BFS; not always feasible