CS540 Introduction to Artificial Intelligence
Lecture 15

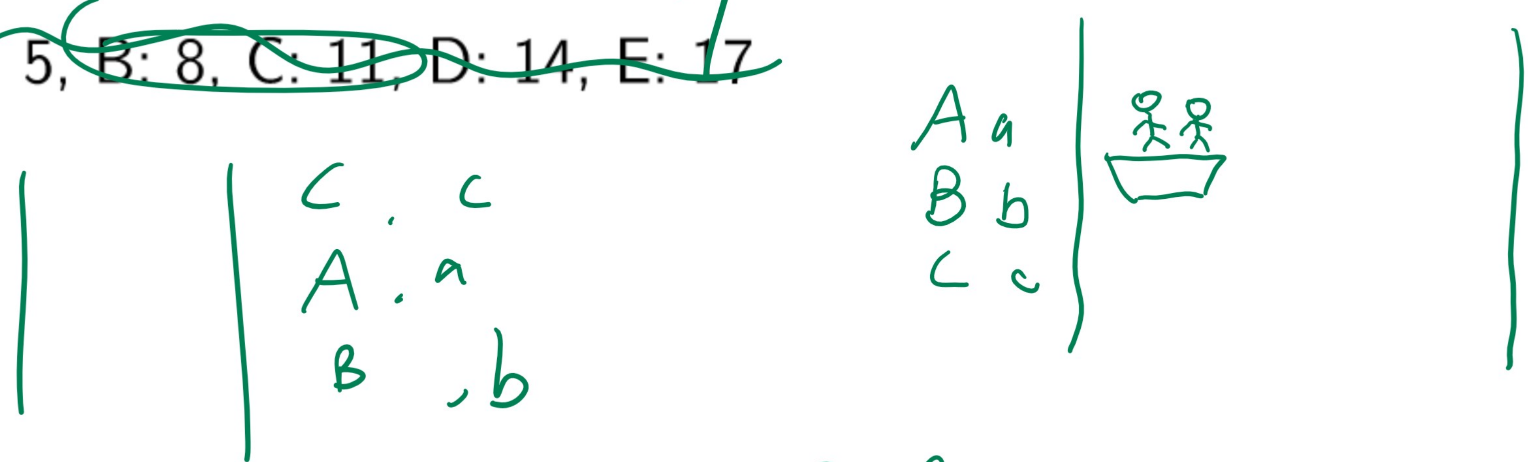
Young Wu
Based on lecture slides by Jerry Zhu and Yingyu Liang

July 15, 2019

River Corssing Problem

Quiz (Participation)

- Three married couples need to cross the river. The boat holds no more than two people. No woman can be in the presence of another man unless her husband is also present. What is the minimum number of times the boat needs to go across the river?
- A: 5, B: 8, C: 11, D: 14, E: 17



Learning vs Search

Motivation

- In reinforcement learning, the reward and state transition need to be learned by taking actions.
- In search problems, the reward and state transitions are given.
- The problem is to find a sequence of actions that lead to the goal with minimum cost.

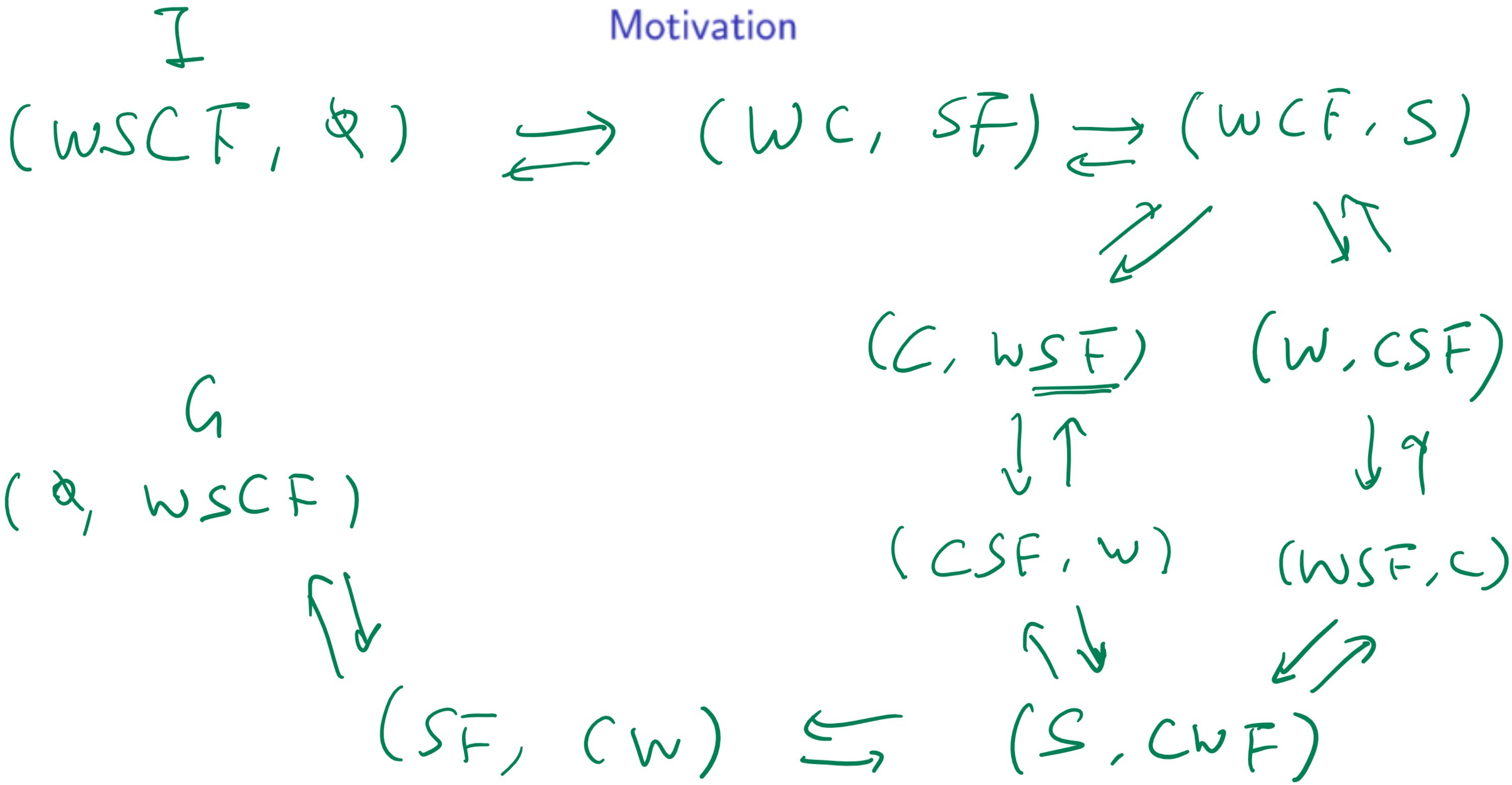
Search Problem Applications

Motivation

- Puzzles and games.
- Navigation: route finding.
- Motion planning.
- Scheduling.

Wolf, Sheep, Cabbage Example

Motivation



Search Problem

Motivation

- State space S is the set of all valid configurations.
- Initial states I and goal states G are subsets of S.
- Successor function $s'(s)$ given the current state s is the set of states reachable in one step from s.
- There is a cost (or negative reward) associated with moving from s to $s'(s)$.
- The search problem is the problem of finding a solution path from a state in I to a state in G, usually with minimum total cost.

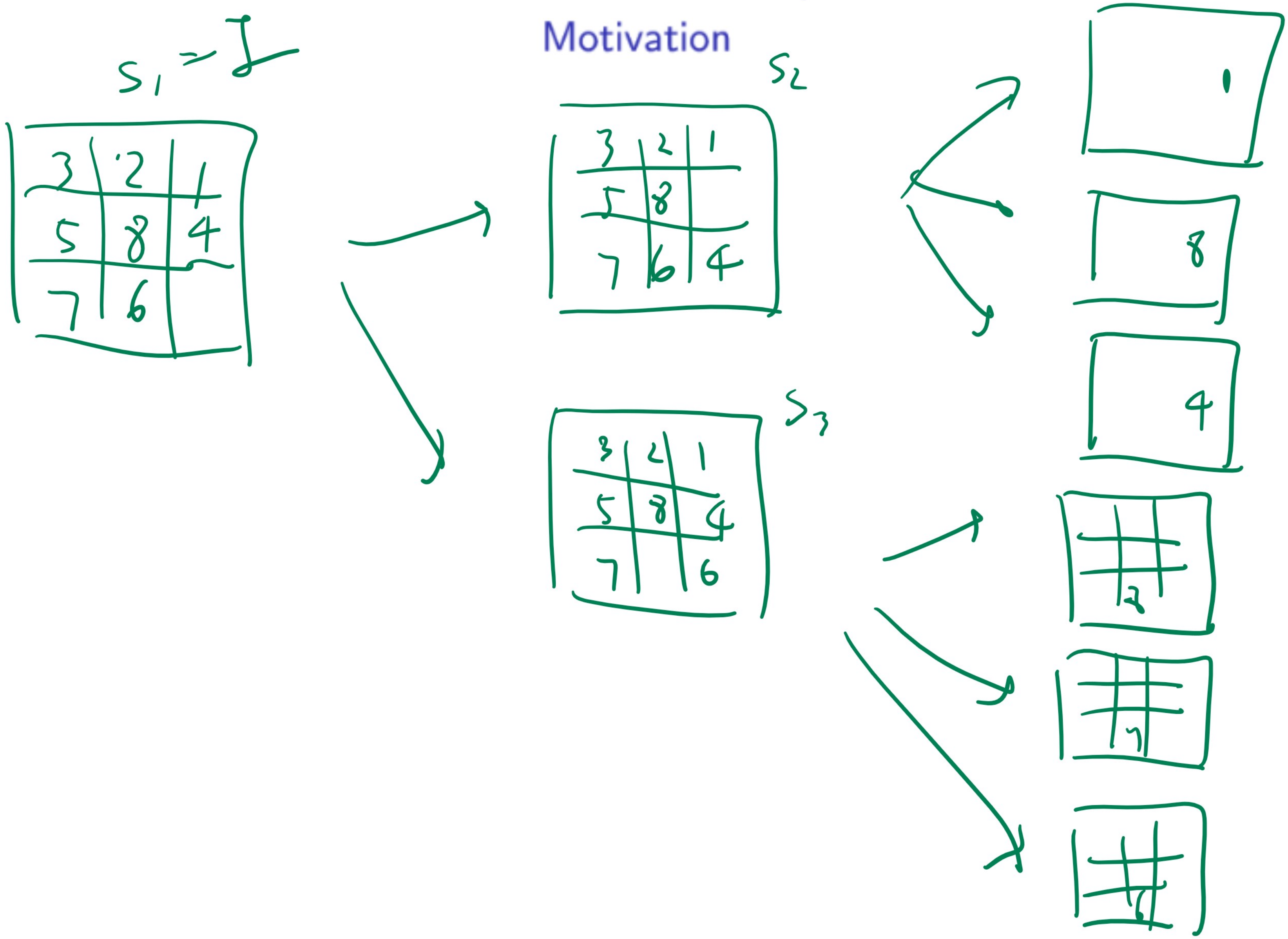
State Space

Motivation

- The states need to represent all necessary information about the game.
- The actions are discrete and deterministic and are determined by the successor function.
- Each possible action at state s is associated with a state in the set $s'(s)$.

8 Puzzle Example

Motivation



Sizes of State Space

Motivation

- Tic Tac Toe: 10^3
- Checkers: 10^{20}
- Chess: 10^{50}
- Go: 10^{170}

State Space Graph

Definition

- A state space can be represented by a weighted directed graph (V, E, c).
- V is the set of vertices (also called nodes).
- E is the set of edges (also called arcs). Each edge is directed from one vertex to another vertex and represents an action.
- c is the cost (also called weights) associated with each edge. The costs are positive.

Search Problem on Graph

Definition

- Search starts at an initial state and finishes if one of the goal states is reached.
- The solution is a path in the graph from an initial state to a goal state.
- The cost of a solution is the sum of edge costs on the solution path.
- The optimal solution is the solution with the lowest cost.

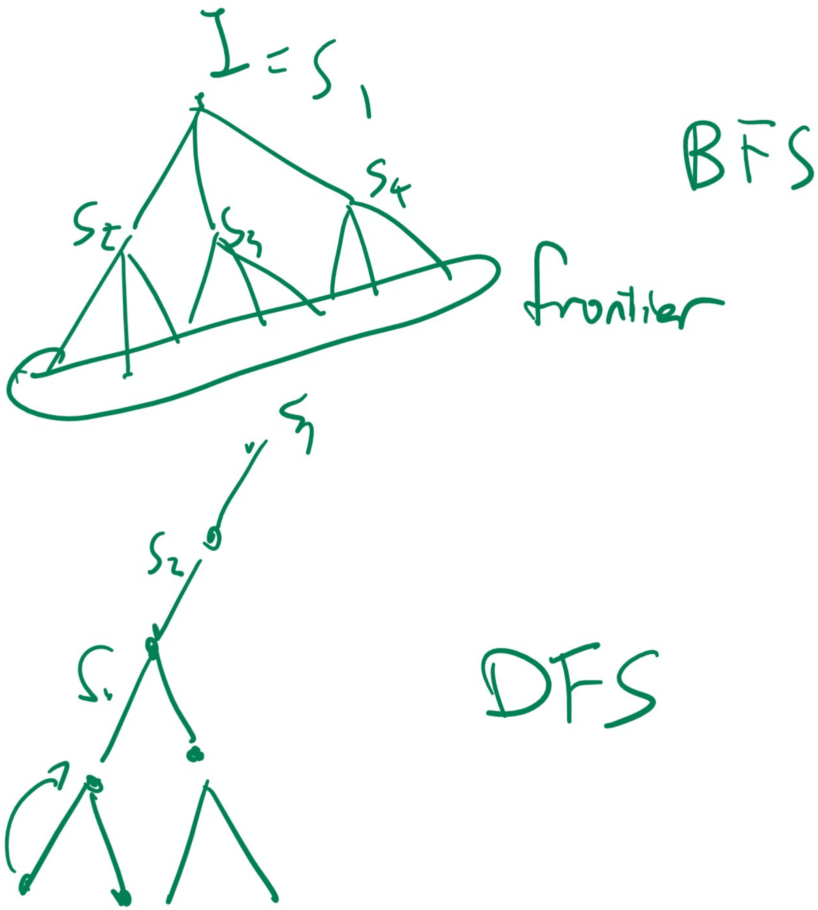
Expansion

Definition

- Vertices that are explored so far are stored in a tree called the state space search tree.
- Expanding a vertex means to generate all successor vertices and add them (and the associated edges) to the state space search tree.
- The leaves of the search tree are unexpanded and are called the frontier (sometimes called the fringe).
- The search strategies differ in the order in which the vertices are expanded.

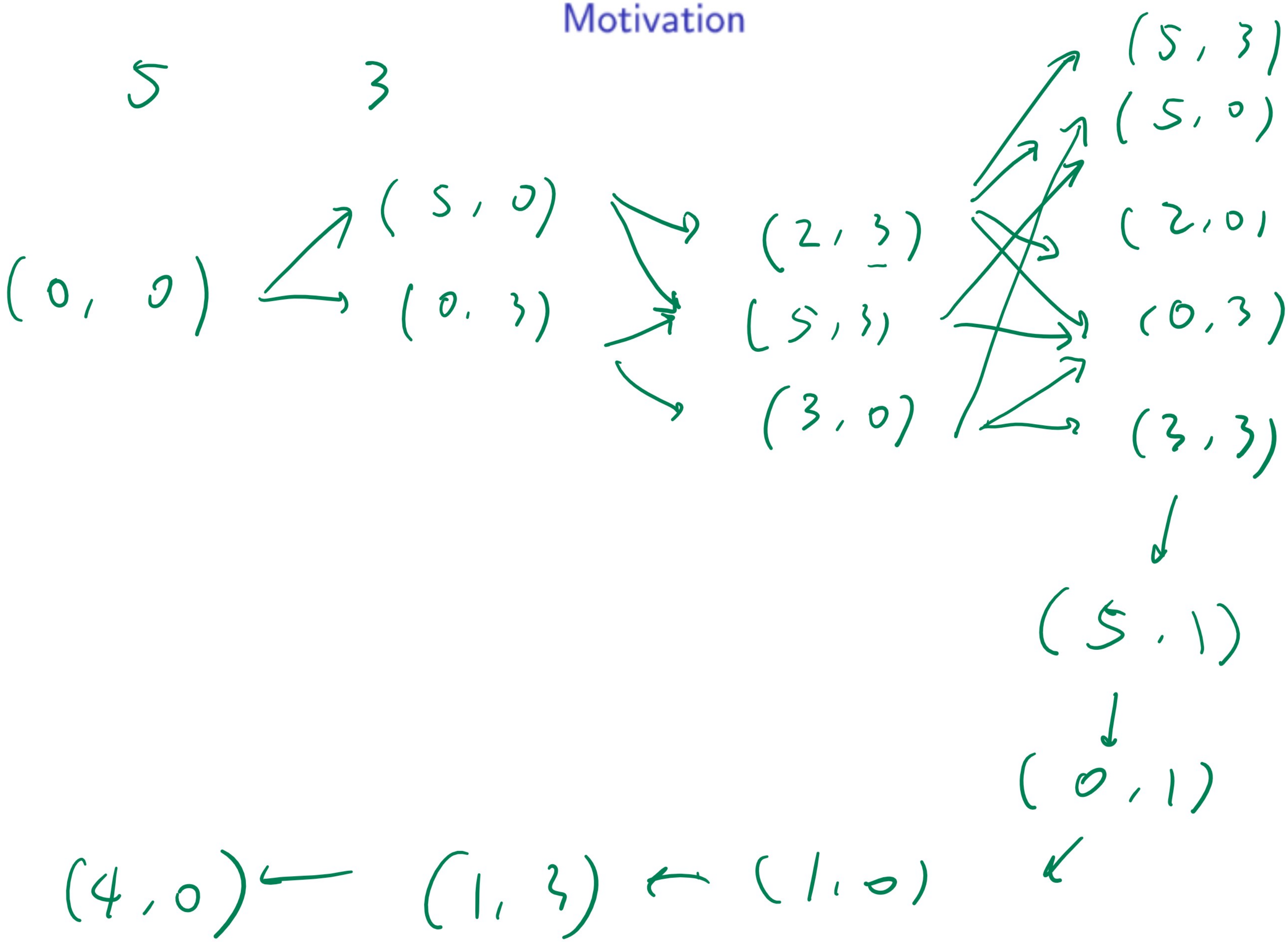
Water Jugs Example, Part I

Motivation



Water Jugs Example, Part II

Motivation



Performance

Definition

- A search strategy is complete if it finds at least one solution.
- A search strategy is optimal if it finds the optimal solution.
- For uninformed search, the costs are assumed to be 1 for all edges $c = 1$.

Complexity

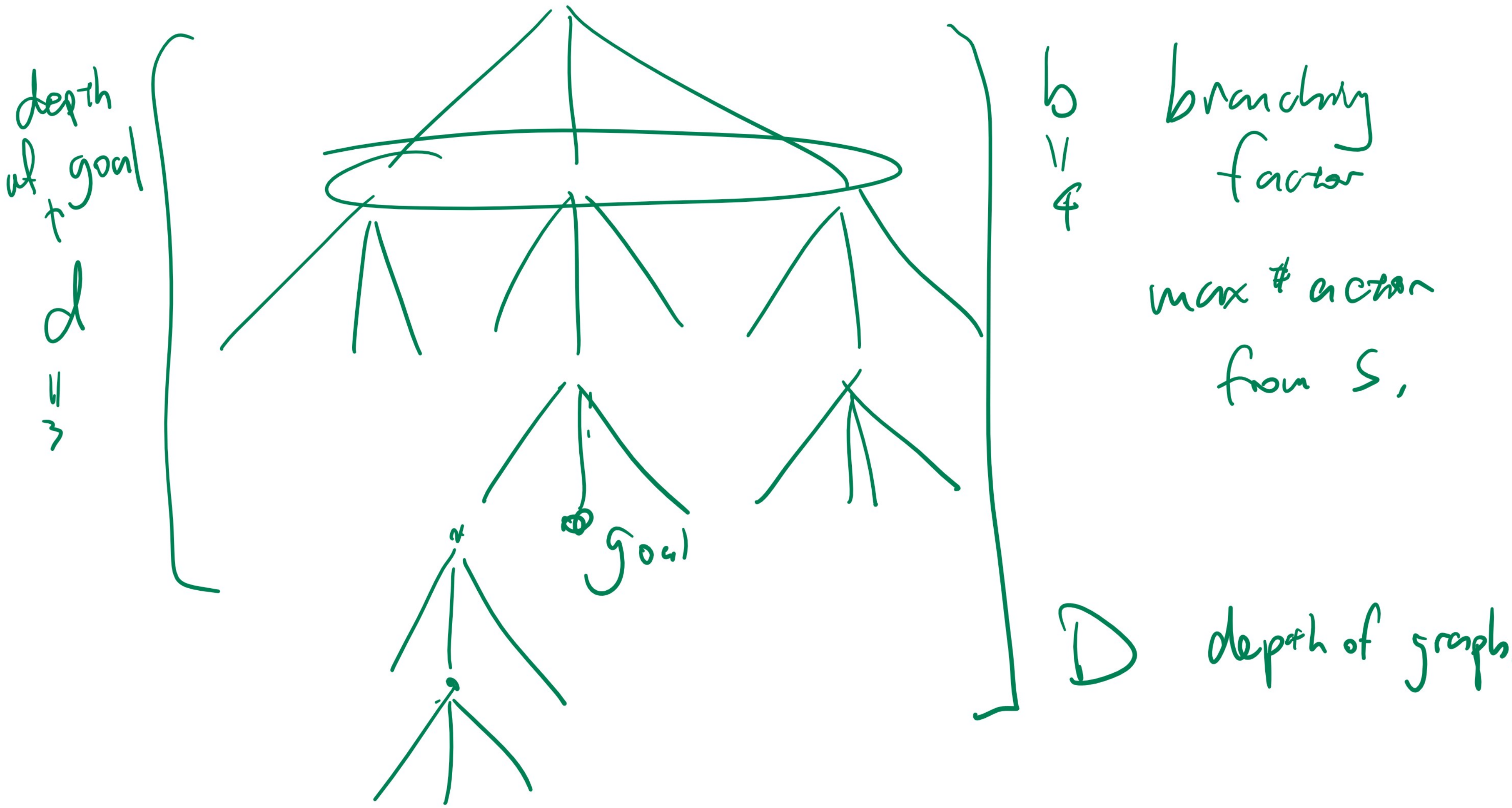
Definition

- The time complexity of a search strategy is the worst case maximum number of vertices expanded.
- The space complexity of a search strategy is the worst case maximum number of states stored in the frontier at a single time.
- Notation: the goals are d edges away from the initial state. This means assuming a constant cost of 1, the optimal solution has cost d. The maximum depth of the graph is D.
- Notation: the branching factor is b, the maximum number of actions associated with a state.

$$b = \max_{s \in V} \left| s'(s) \right|$$

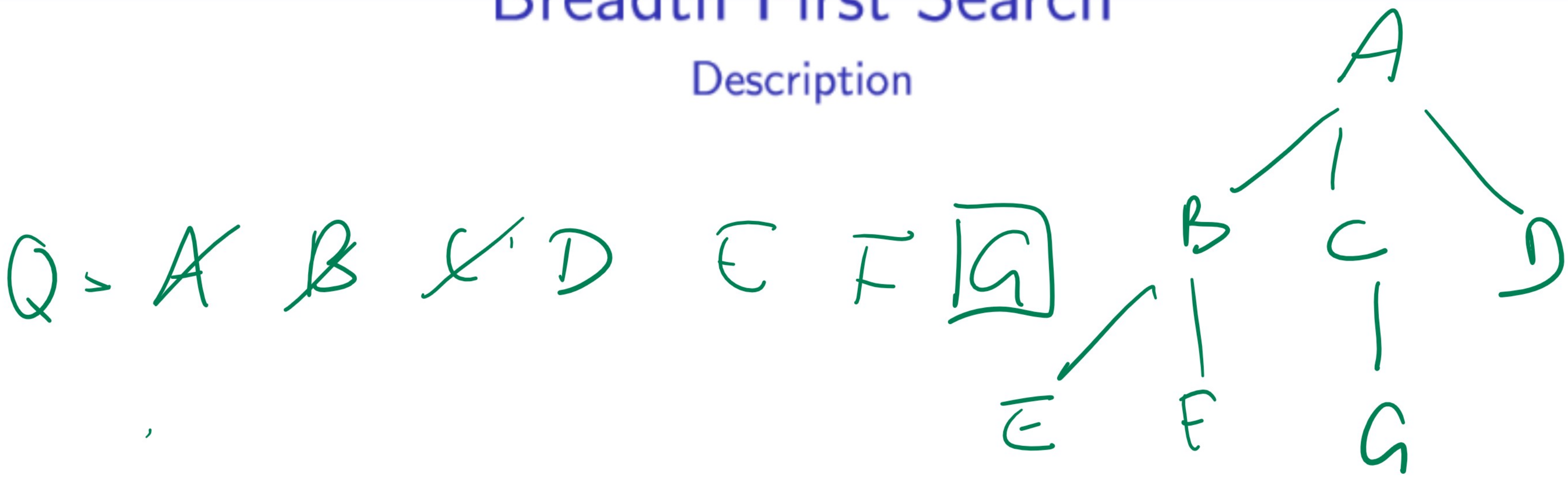
Search Tree Diagram

Definition



Breadth First Search

Description

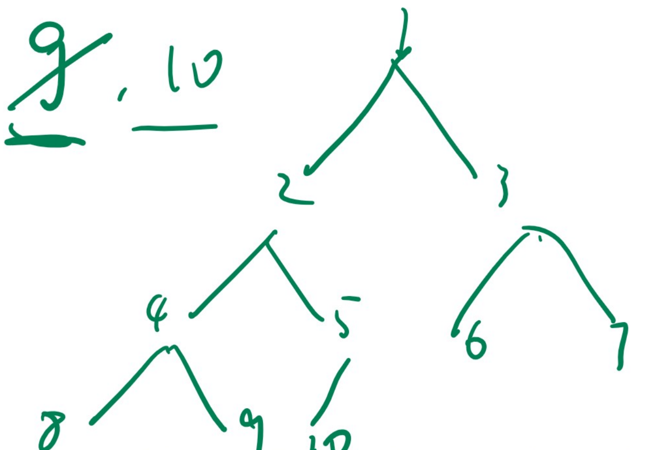


- Use Queue (FIFO) for the frontier.
- Remove from the front, add to the back.

BFS Example

Quiz (Graded)

- Fall 2018 Midterm Q2, Fall 2017 Midterm Q13, Fall 2010 Final Q2
- Suppose the states are positive integers between 1 and 10, initial state is 1, goal state is 9, successors of i is $2i$ and $2i+1$ (if exist). What a BFS expansion sequence?
- A: 1, 2, 3, 4, 5, 6, 7, 8, 9
- B: 1, 2, 4, 8, 3, 5, 7, 9
- C: 1, 2, 4, 8, 9
- D: 1, 2, 3, 2, 4, 5, 3, 6, 7, 2, 4, 8, 9
- E: 1, 2, 3, 2, 4, 2, 5, 3, 6, 3, 7, 2, 4, 8, 2, 4, 9



Maze BFS Example

Motivation

Breadth First Search

Algorithm

- Input: a weighted digraph (V, E, c), initial states I and goal states G.
- Output: a path from I to G.
- EnQueue initial states.

$$Q = I$$

- While Q is not empty and goal is not ~~fond~~ deQed, deQueue Q and enQueue its successors.

$$s = Q_0$$
$$Q = Q + s'(s)$$

Breadth First Search Performance

Discussion

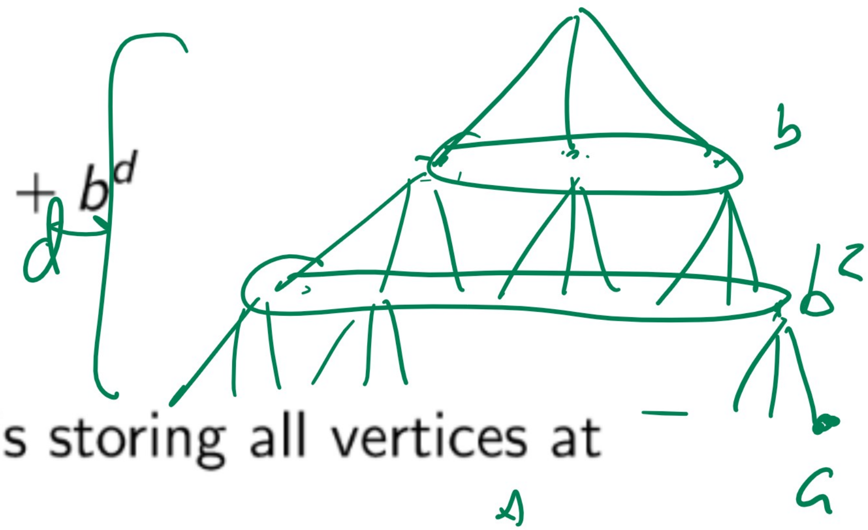
- BFS is complete.
- BFS is optimal with $c = 1$.

Breadth First Search Complexity

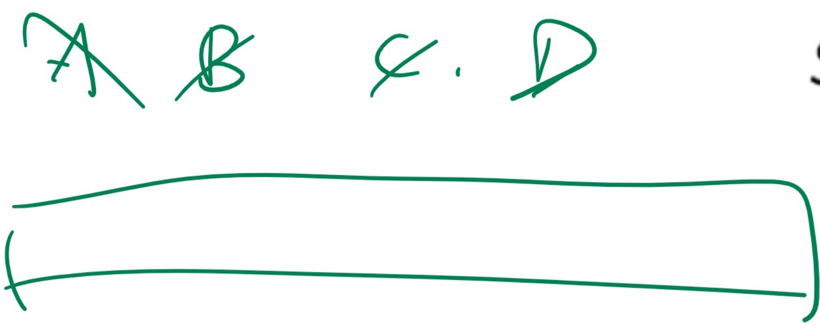
Discussion

- Time complexity: the worst case occurs when the goal is the last vertex at depth d.

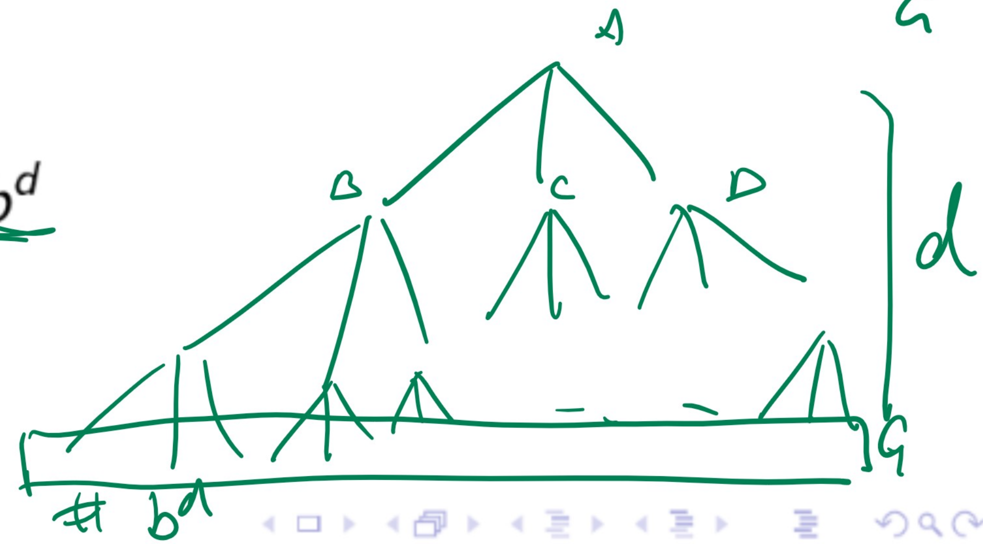
$$T = b + b^2 + \ldots + b^d$$



- Space complexity: the worst case is storing all vertices at depth d is in the frontier.

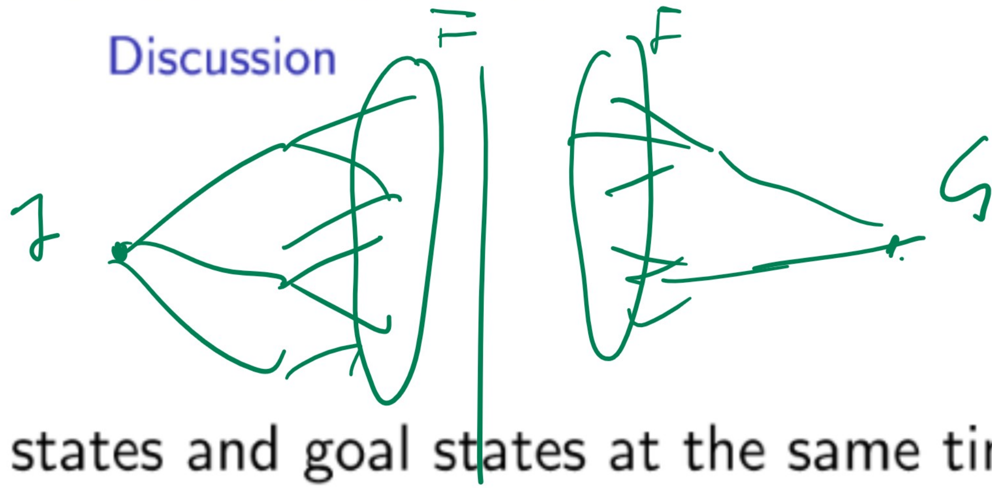


$$S = b^d$$



BiDirectional Search

Discussion

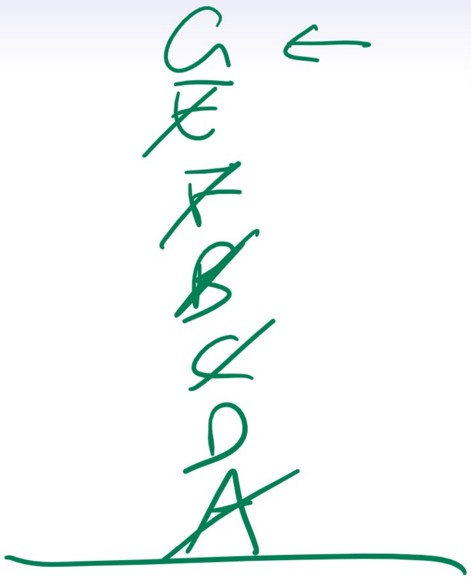


- BFS from the initial states and goal states at the same time.
- The search stops when the two frontiers meet (have non-empty intersection) in the middle.
- The time and space complexity is the same as BFS with depth $\frac{d}{2}$.

$$T = 2(b + b^2 + \dots + b^{\frac{d}{2}})$$

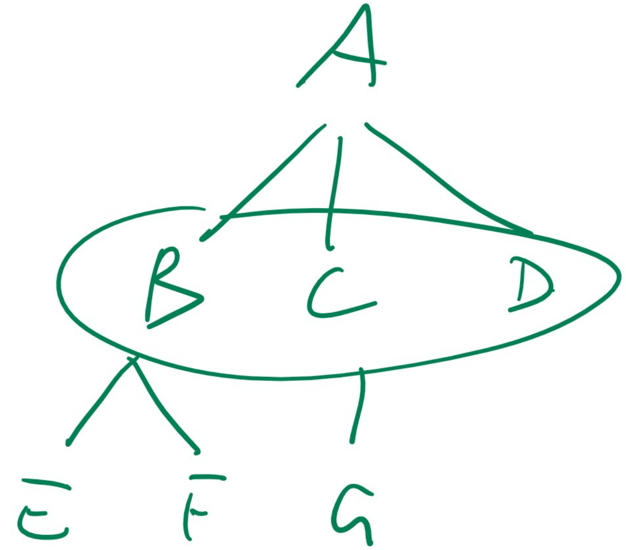
$$S = 2b^{\frac{d}{2}}$$

$$Q(I) \cap Q(G) \neq \emptyset \rightarrow \text{meet.}$$



Depth First Search

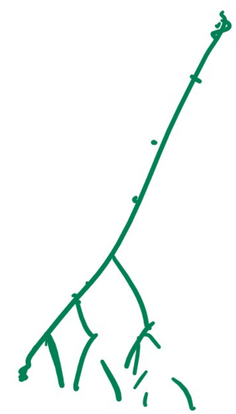
Description



- Use Stack (LIFO) for the frontier.
- Remove from the front, add to the front.

A B E F C G

list of expanded vertices.



DFS Example

Quiz (Graded)

- Fall 2018 Midterm Q2, Fall 2017 Midterm Q13, Fall 2010 Final Q2
- Suppose the states are positive integers between 1 and 10, initial state is 1, goal state is 9, successors of i is $2i$ and $2i + 1$ (if exist). What a DFS expansion sequence?
- A: 1, 2, 3, 4, 5, 6, 7, 8, 9
- B: 1, 2, 4, 8, 3, 5, 7, 9
- C: 1, 2, 4, 8, 9
- D: 1, 2, 3, 2, 4, 5, 3, 6, 7, 2, 4, 8, 9
- E: 1, 2, 3, 2, 4, 2, 5, 3, 6, 3, 7, 2, 4, 8, 2, 4, 9

Maze DFS Example

Motivation

Depth First Search

Algorithm

- Input: a weighted digraph (V, E, c), initial states I and goal states G.
- Output: a path from I to G.
- Push initial states.

$$S = I$$

- While S is not empty and goal is not ~~found~~ popped, pop S and push its successors.

$$s = S_0$$
$$S = s'(s) + S$$

Depth First Search Performance

Discussion

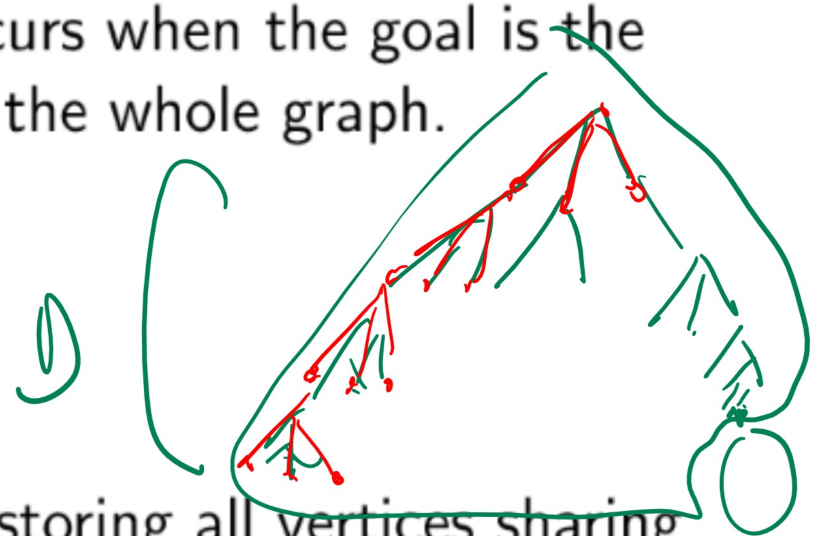
- DFS is incomplete if $D = \infty$.
- DFS is not optimal.

Depth First Search Complexity

Discussion

- Time complexity: the worst case occurs when the goal is the root of the last subtree expanded in the whole graph.

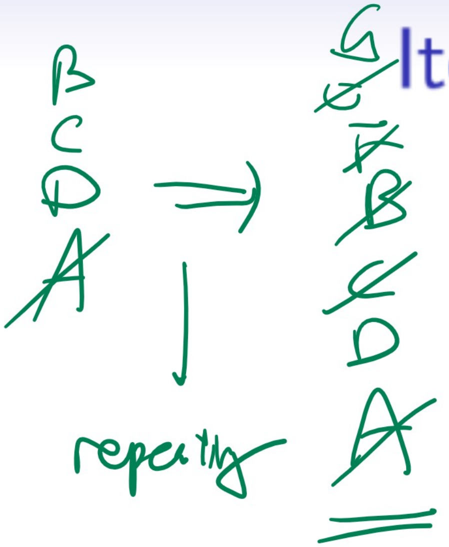
$$T \approx b^D$$



- Space complexity: the worst case is storing all vertices sharing the parents with vertices in the current path.

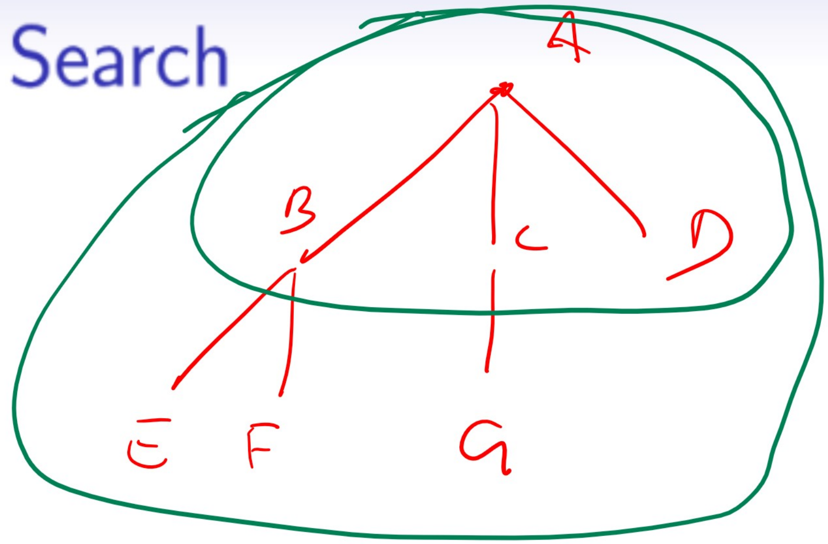
$$S = (b-1)\,D$$

	BFS	DFS
Time	small	large
Space	large	small



Iterative Deepening Search

Discription

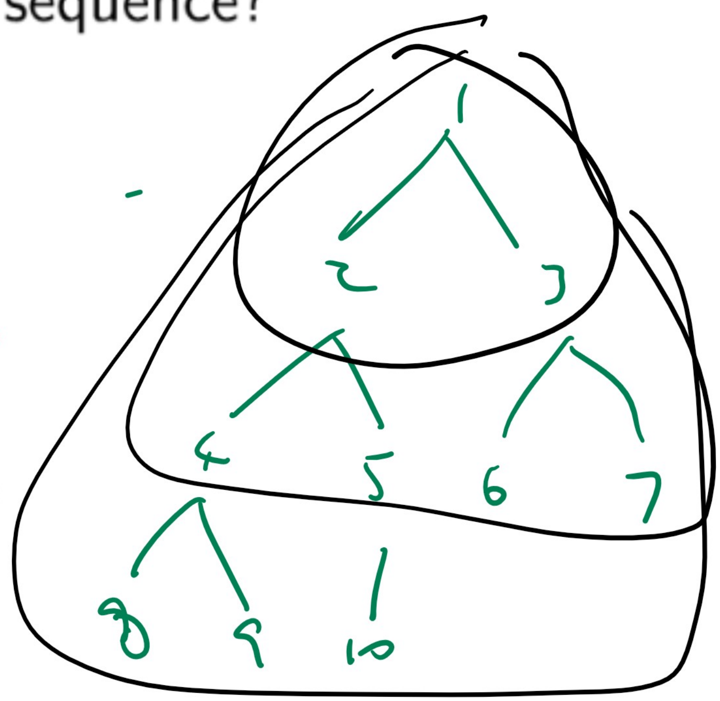


- DFS but stop if path length > 1
- repeat DFS but stop if path length > 2
- ...
- repeat DFS but stop if path length $> d$

IDS Example

Quiz (Graded)

- Fall 2018 Midterm Q2, Fall 2017 Midterm Q13, Fall 2010 Final Q2
- Suppose the states are positive integers between 1 and 10, initial state is 1, goal state is 9, successors of i is $2i$ and $2i + 1$ (if exist). What a IDS expansion sequence?
- A: 1, 2, 3, 4, 5, 6, 7, 8, 9
- B: 1, 2, 4, 8, 3, 5, 7, 9
- C: 1, 2, 4, 8, 9
- D: 1, 2, 3, 2, 4, 5, 3, 6, 7, 2, 4, 8, 9
- E: 1, 2, 3, 2, 4, 2, 5, 3, 6, 3, 7, 2, 4, 8, 2, 4, 9



1, 2, 3, 1 2 4 5 3 6 7

1, 2, 4, 8, 9

Maze IDS Example

Motivation

Iterative Deepening Search

Algorithm

- Input: a weighted digraph (V, E, c), initial states I and goal states G.
- Output: a path from I to G.
- Perform DFS on the digraph restricted to vertices with depth $\leqslant 1$ from the initial state.
- Perform DFS on the digraph restricted to vertices with depth $\leqslant 2$ from the initial state.
- Repeat until the goal is ~~found.~~ deQed.

Iterative Deepening Search Performance

Discussion

- IDS is complete.
- IDS is optimal with $c = 1$.

Iterative Deepening Search Complexity

Discussion

- Time complexity: the worst case occurs when the goal is the last vertex at depth d.

$$T = db + (d-1)\,b^2 + \ldots + 3b^{d-1} + 2b^{d-1} + 1b^d$$

= $O(b^d)$ same as BFS

- Space complexity: it has the same space complexity as DFS.

$$S = (b-1)\,D$$

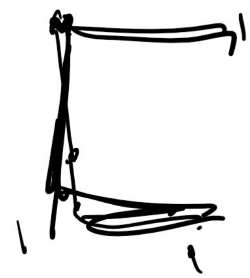
Non-Tree Search

Discussion

- If the state space is not a tree, search strategies need to remember the states that are already expanded.
- A vertex should be removed from the frontier if it is already expanded.

Configuration Space

Discussion



Uniformed vs Informed Search

Discussion

- Uninformed search means only the goal G and the successor function s' are given.
- Informed search means which non-goal states are better is also known.
- Usually, iterative deepening is used for uninformed search.