Autonomous Robotics

Planning

Josiah Hanna
University of Wisconsin — Madison

Announcements

Midterm **TONIGHT in CS 1221.**

Grading: HW 2 is underway, everything else has been graded and returned to you.

Midterm survey. Please complete ASAP!

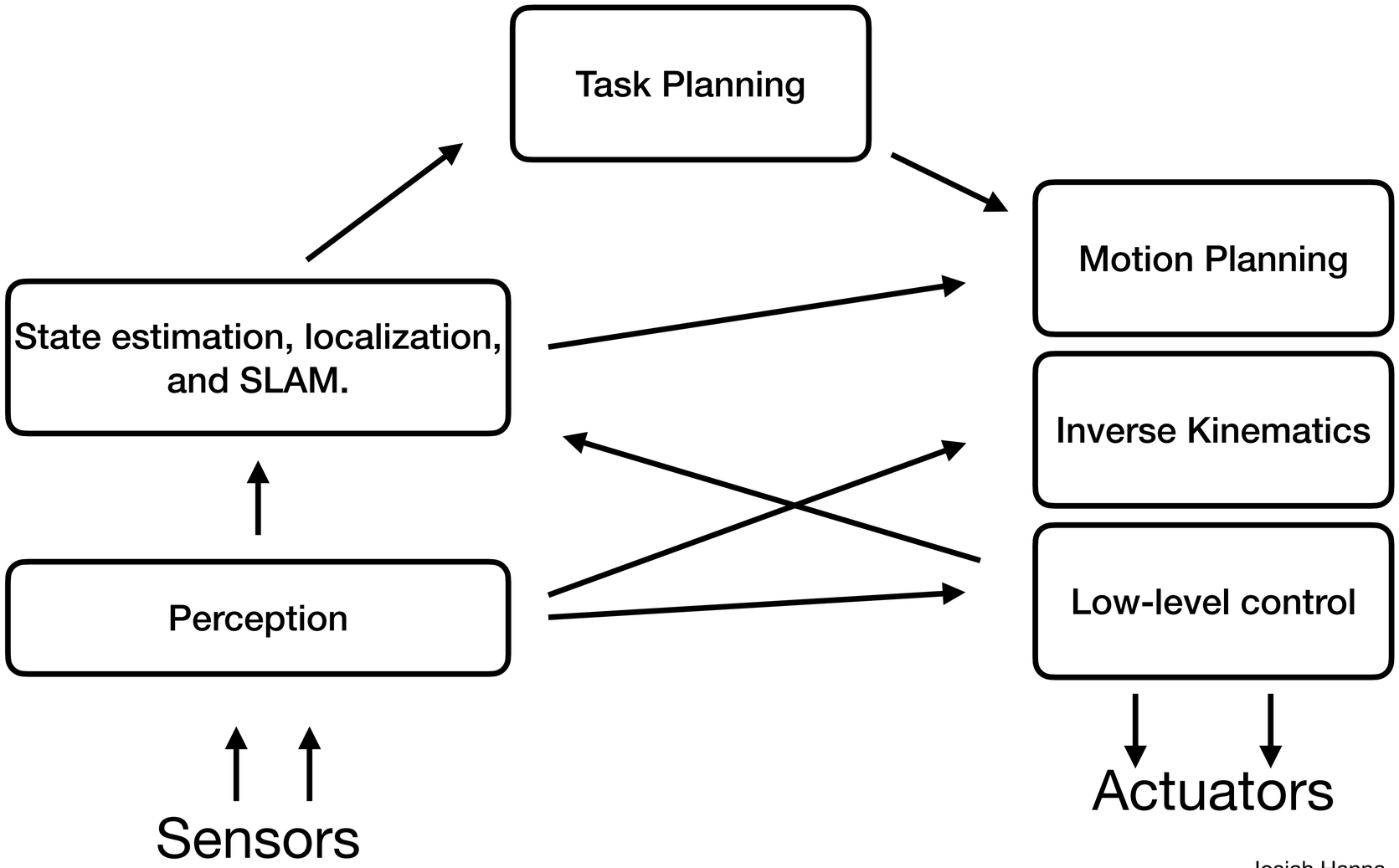
Office hours: today at 1pm

Learning Outcomes

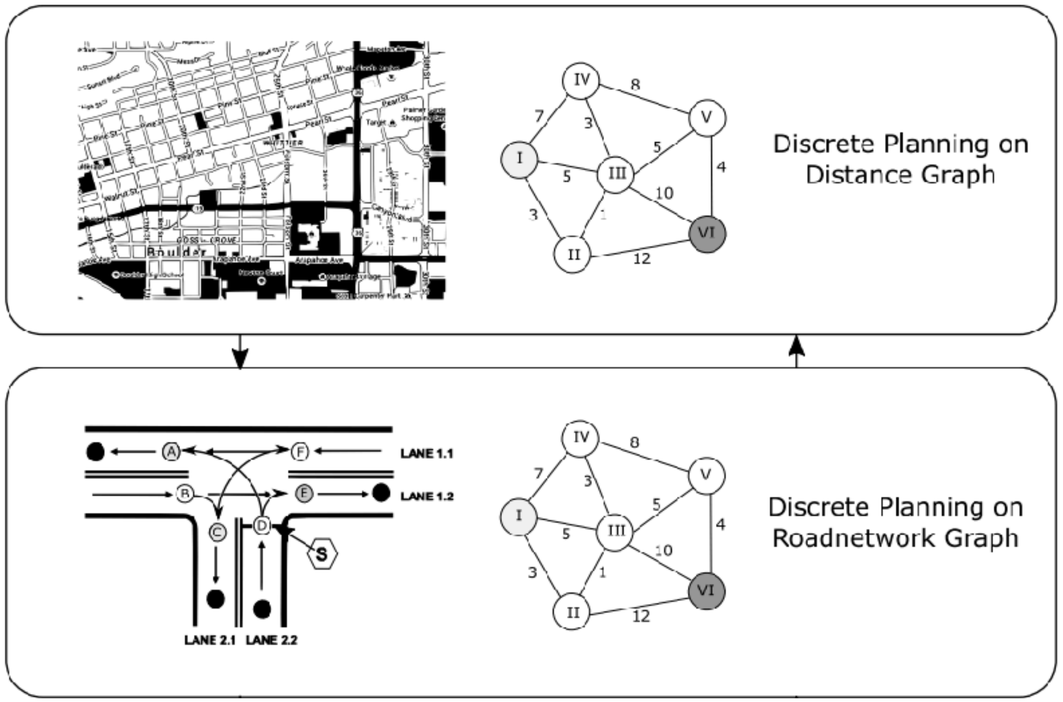
After today's lecture, you will:

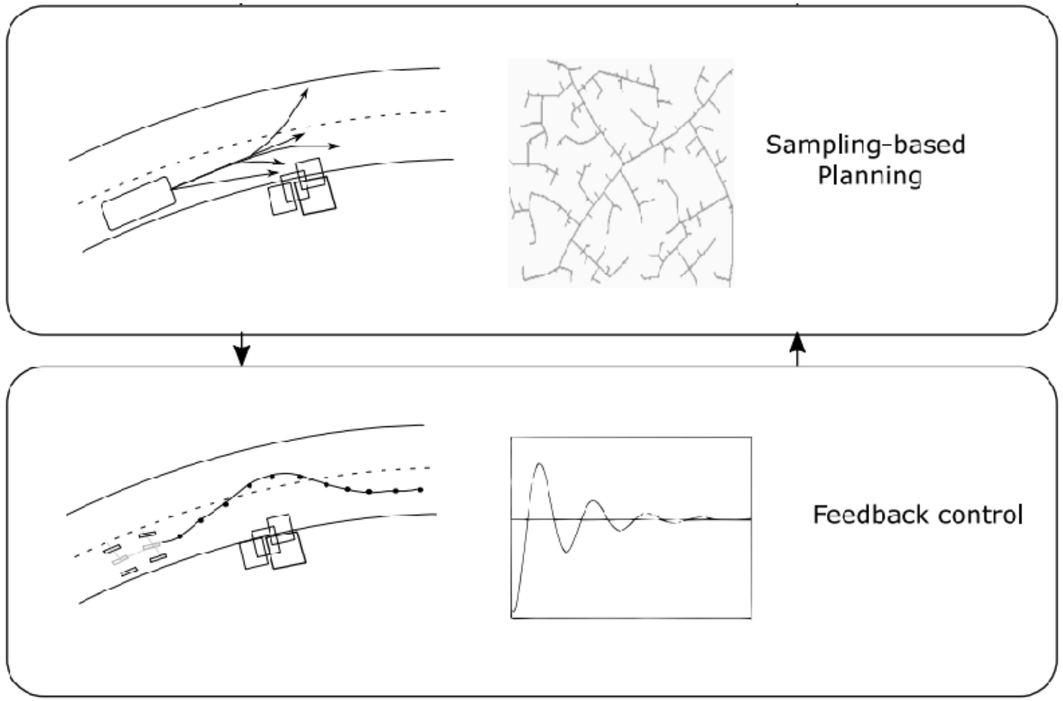
- Be able to formulate planning problems in robotics.
- Be able to identify key algorithms for graph-based planning.
- Understand strengths and weaknesses of graph-based planning in robotics.

Components so far



Motion Control

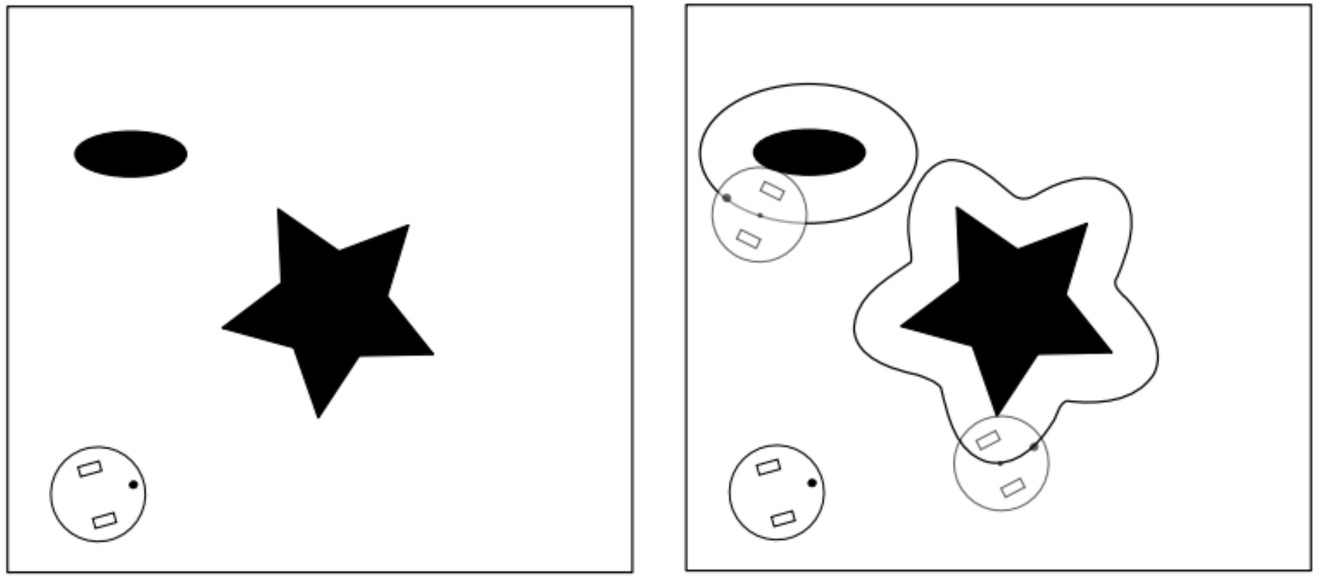




Path Planning

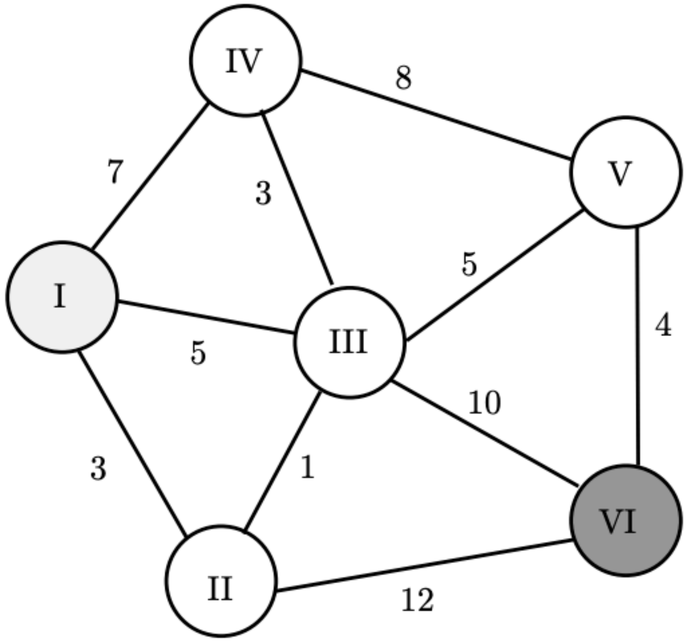
- Find a sequence of states that lead from an initial state to a goal state.
- Typically want the shortest or lowest cost path among all valid paths.
- Why useful?

Configuration Space



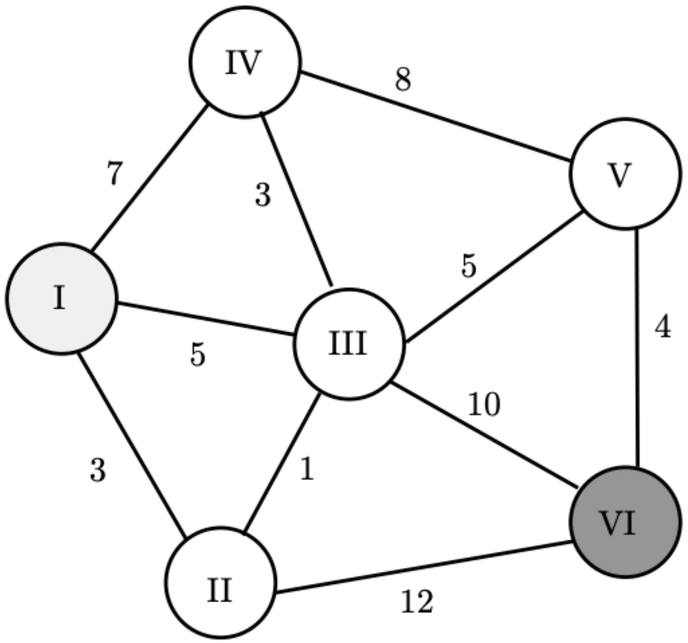
Graph-based Planning

- Properties: completeness, optimality, space & time complexity.



General Structure

- Start at one state (usually the start state).
- Add neighbors of the start state to some data structure, f.
- while f is not empty:
  - Remove a state, n, from f.
  - Check if n is the goal state.
  - If not, add the neighbors of n to f.



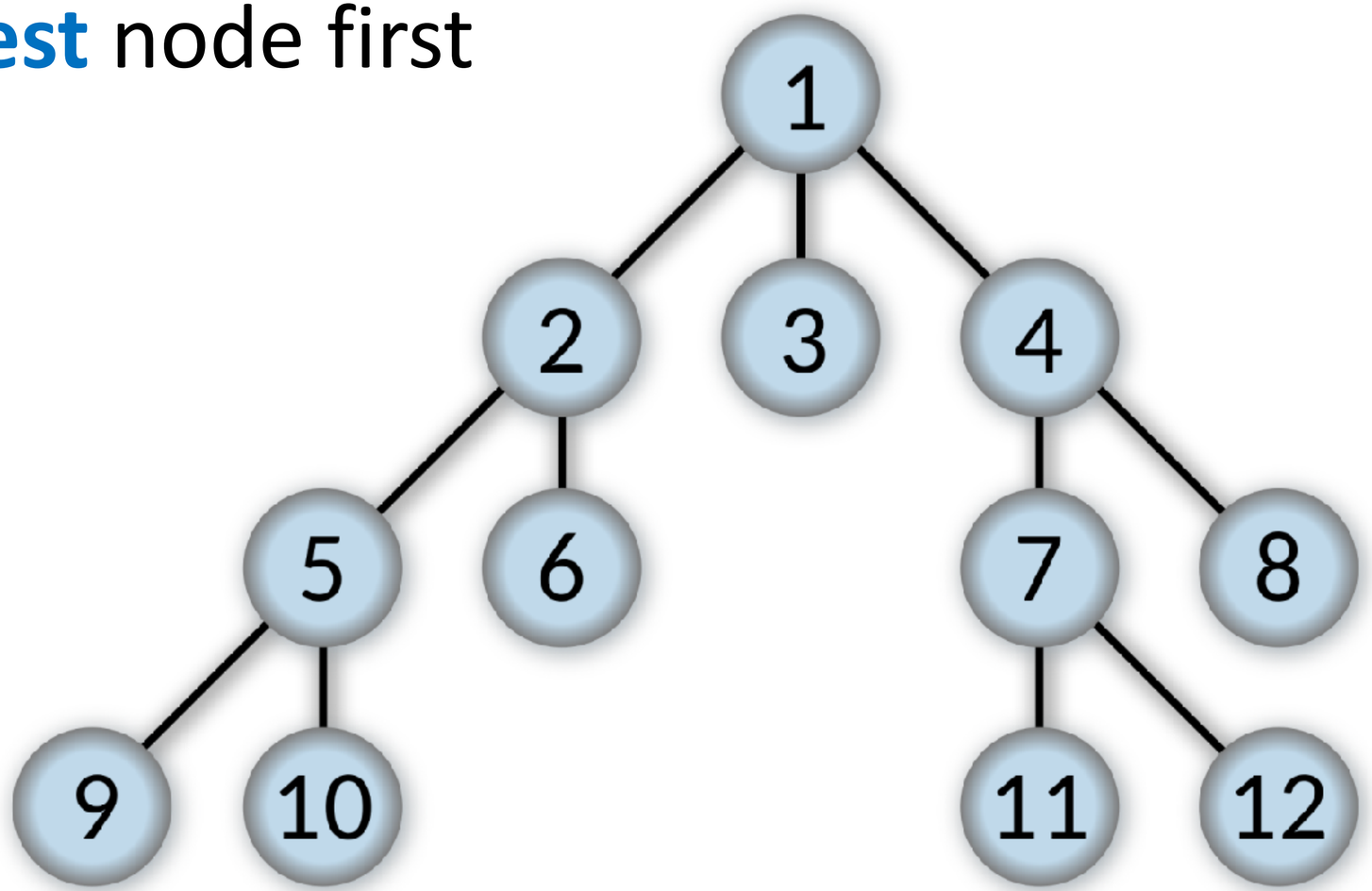
Breadth-First Search

Recall: expand **shallowest** node first

- Data structure: queue
- **Properties**:
  - Complete
  - Optimal (if edge cost 1)
  - Time O(b^d)
  - Space O(b^d)

Depth

Branching Factor



Wiki

Uniform Cost Search

Like BFS, but keeps track of cost

- Expand least cost node
- Data structure: priority queue
- **Properties**:
  - Complete
  - Optimal (if weight lower bounded by ε)
  - Time $O(b^{C^*/\varepsilon})$
  - Space $O(b^{C^*/\varepsilon})$

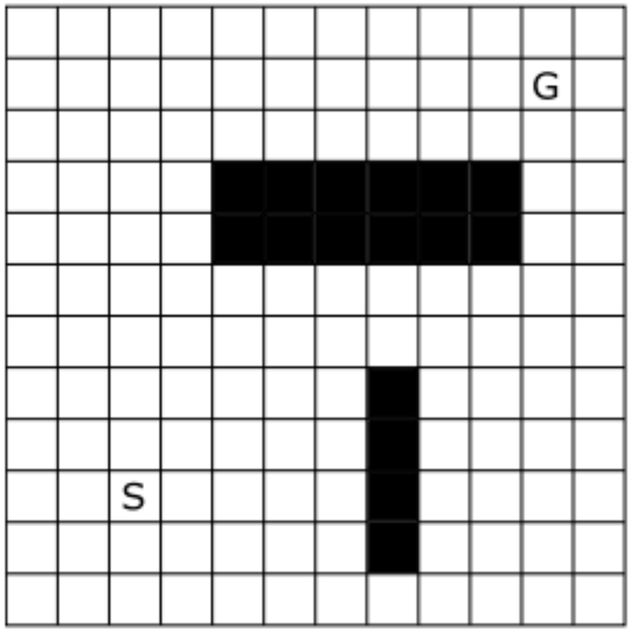
C* is optimal path cost to goal.

ϵ is cost of edge with smallest cost.

Credit: DecorumBY

Djikstra's Algorithm

- AKA Uniform Cost Search

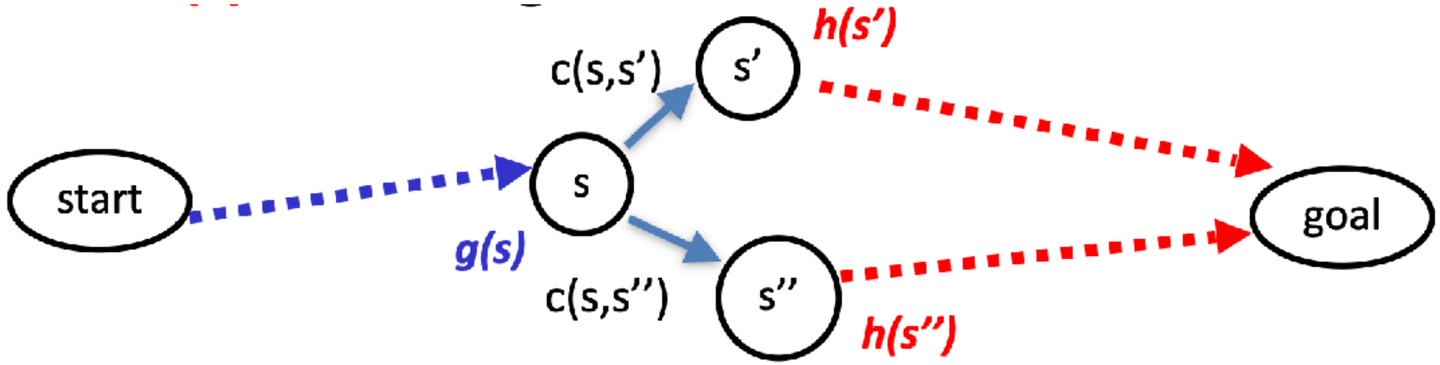


11	10	9	10	11	12						
10	9	8	9	10	11	12				G	
9	8	7	8	9	10	11	12				
8	7	6	7								
7	6	5	6								
6	5	4	5	6	7	8	9	10	11	12	
5	4	3	4	5	6	7	8	9	10	11	12
4	3	2	3	4	5	6		10	11	12	
3	2	1	2	3	4	5		11	12		
2	1	S	1	2	3	4		10	11	12	
3	2	1	2	3	4	5		9	10	11	12
4	3	2	3	4	5	6	7	8	9	10	11

11	10	9	10	11	12	13	14	15	16		
10	9	8	9	10	11	12	13	14	15	G	
9	8	7	8	9	10	11	12	13	14	15	
8	7	6	7							14	15
7	6	5	6							13	14
6	5	4	5	6	7	8	9	10	11	12	13
5	4	3	4	5	6	7	8	9	10	11	12
4	3	2	3	4	5	6		10	11	12	13
3	2	1	2	3	4	5		11	12	13	14
2	1	S	1	2	3	4		10	11	12	13
3	2	1	2	3	4	5		9	10	11	12
4	3	2	3	4	5	6	7	8	9	10	11

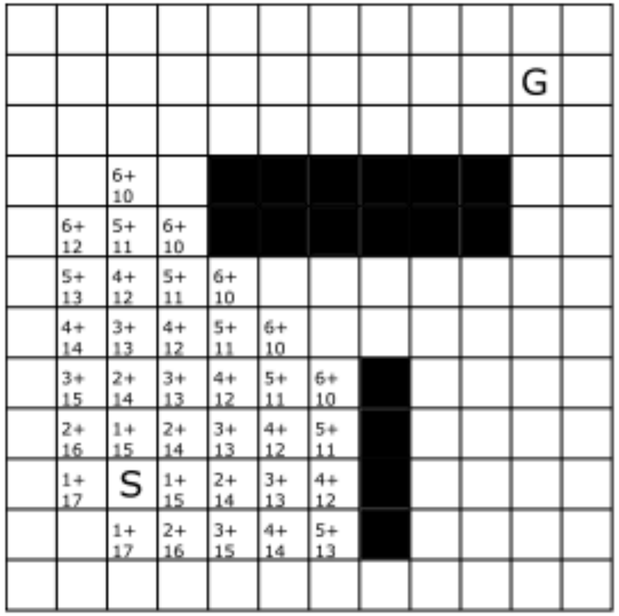
A* Search

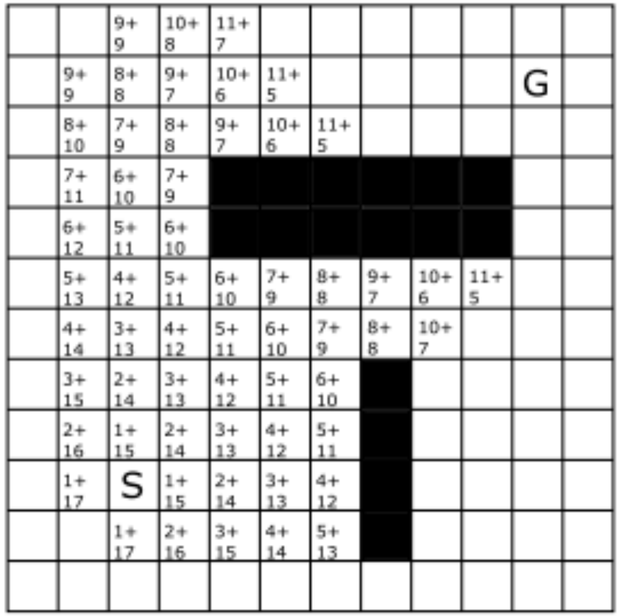
- Use a **heuristic** function to speed-up search.
- Heuristic must be admissible: non-negative and never over-estimates the cost-to-goal.
- A* is uniform cost search with $g(s) + c(s, s') + h(s')$ as the cost for s'.

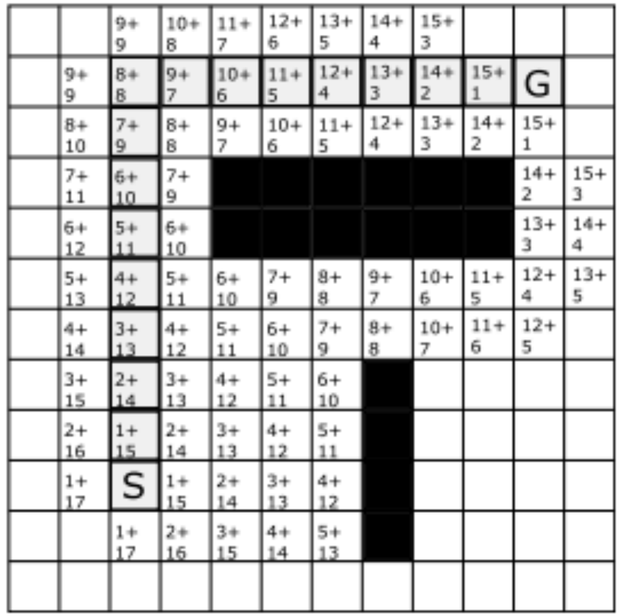


A* Search

- Use a **heuristic** function to speed-up search.

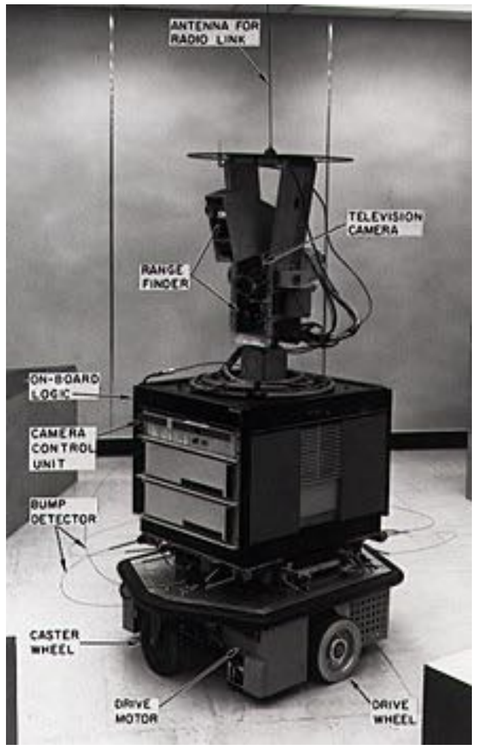






A* Search

Origins: robots and planning



Shakey the Robot, 1960's

Credit: Wiki

Animation: finding a path around obstacle

Credit: Wiki

D* Search

- Search backward from goal to start to compute shortest paths to goal.
- As the robot encounters obstacles, update the path costs.
  - Avoids full re-planning.
  - When is this useful?

Strengths / Weaknesses of Graph Planning

- Strengths:
  - Deterministic and discrete makes it possible to proof properties like completeness.
  - Many robotics planning problems are naturally formulated as discrete planning problems.
- Weaknesses:
  - Need to discretize the state space if continuous. Then only have *resolution completeness.*
  - May require large amounts of memory or large computation time.

Summary

- Discussed graph-based planning
- Introduced several different graph-based planning algorithms.

Action Items

- RL reading due after Spring Break; send a reading response by 12 pm on Monday after SB.

- Midterm: tonight!