ECE 4160/5160

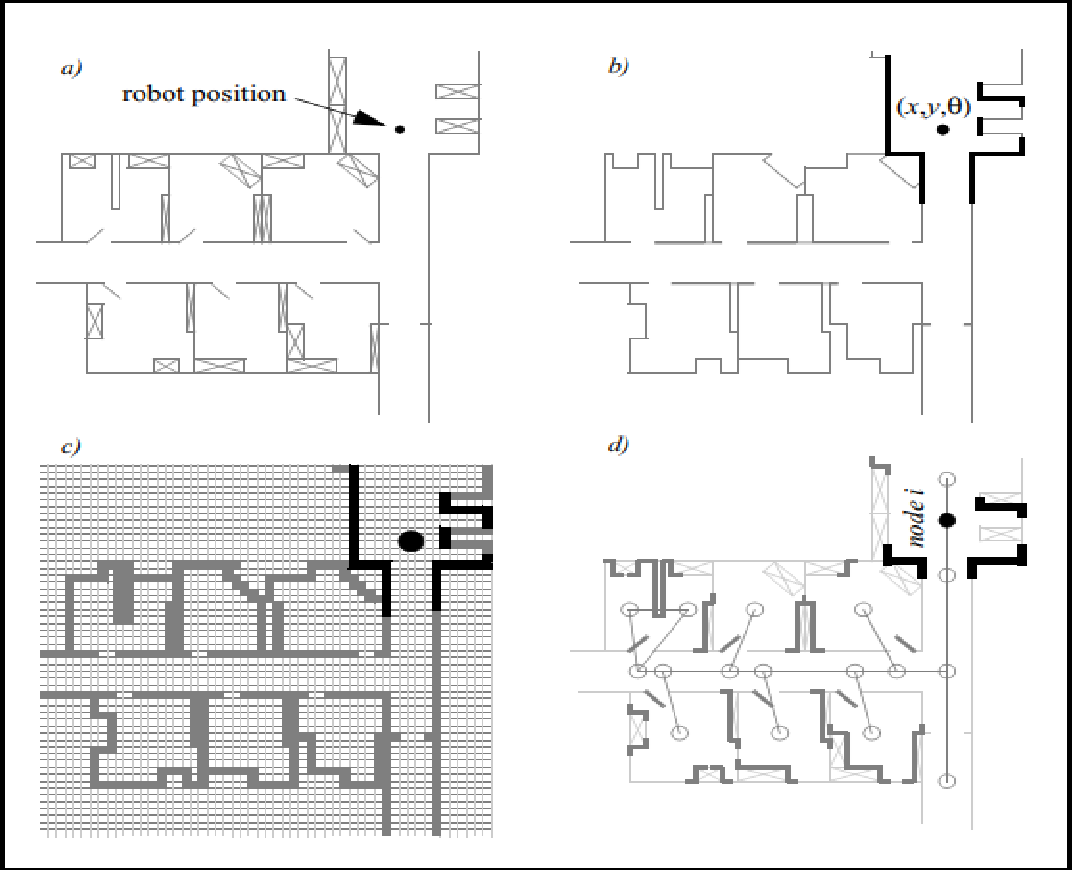
MAE 4910/5910

Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

Map Representations, Graphs and Graph Search

Modelling path planning as a graph search problem

Real world → Configuration Space → Map Representation → Graph Construction → Graph Search

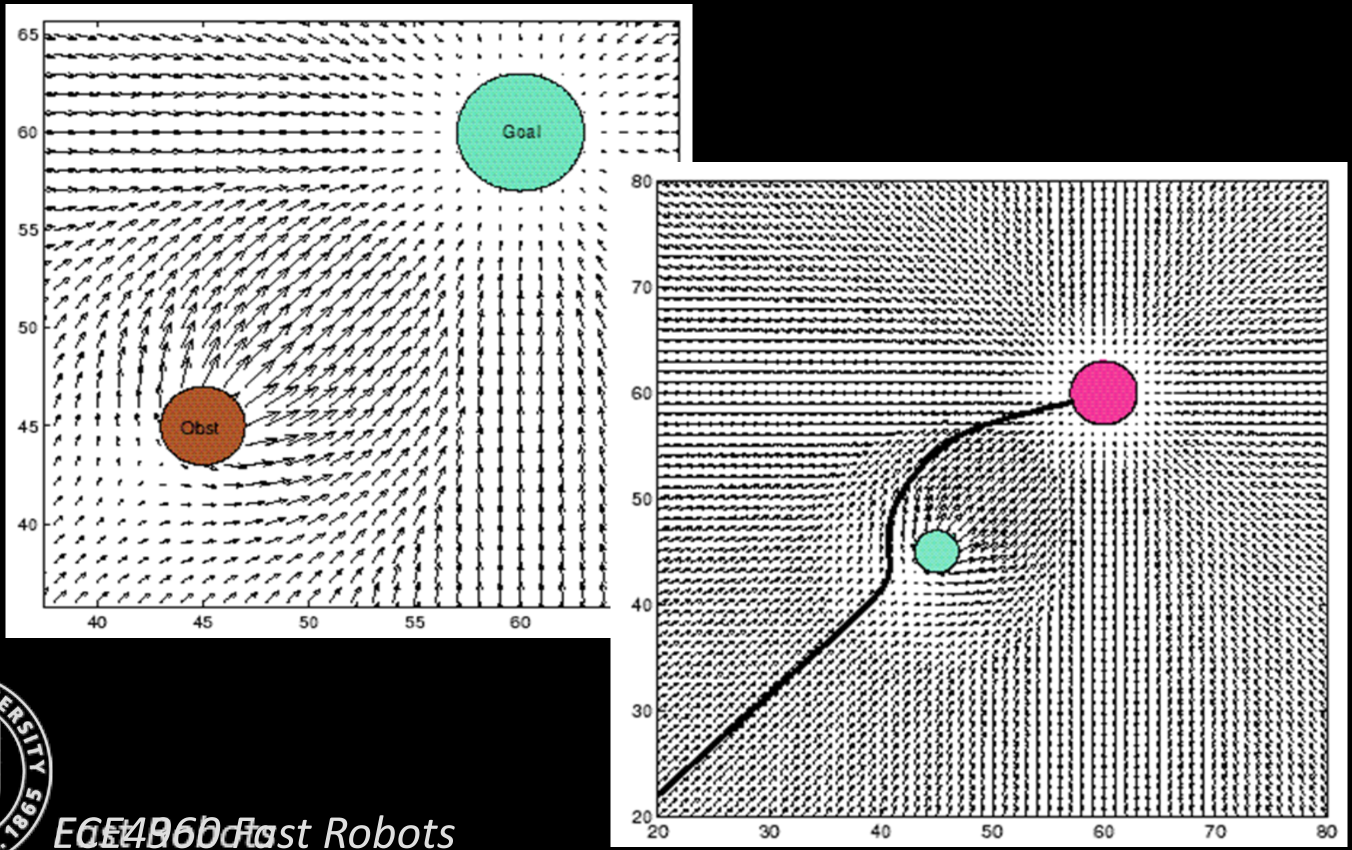


Modelling path planning as a graph search problem

Real world → Configuration Space → Map Representation → Graph Construction → Graph Search

- Topological Graphs
- Cell decomposition
- Visibility Graphs
- RRT
- PRM

Common alternatives
- Optimal control
- Potential fields



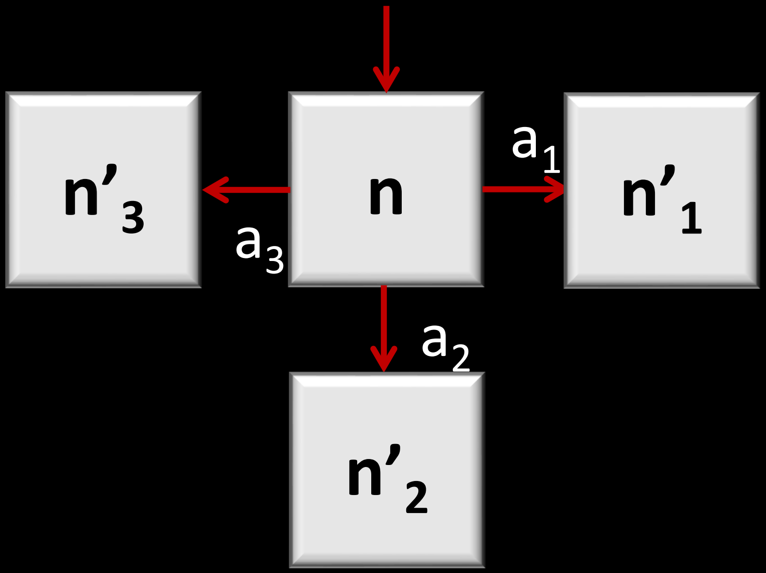
Modelling path planning as a graph search problem

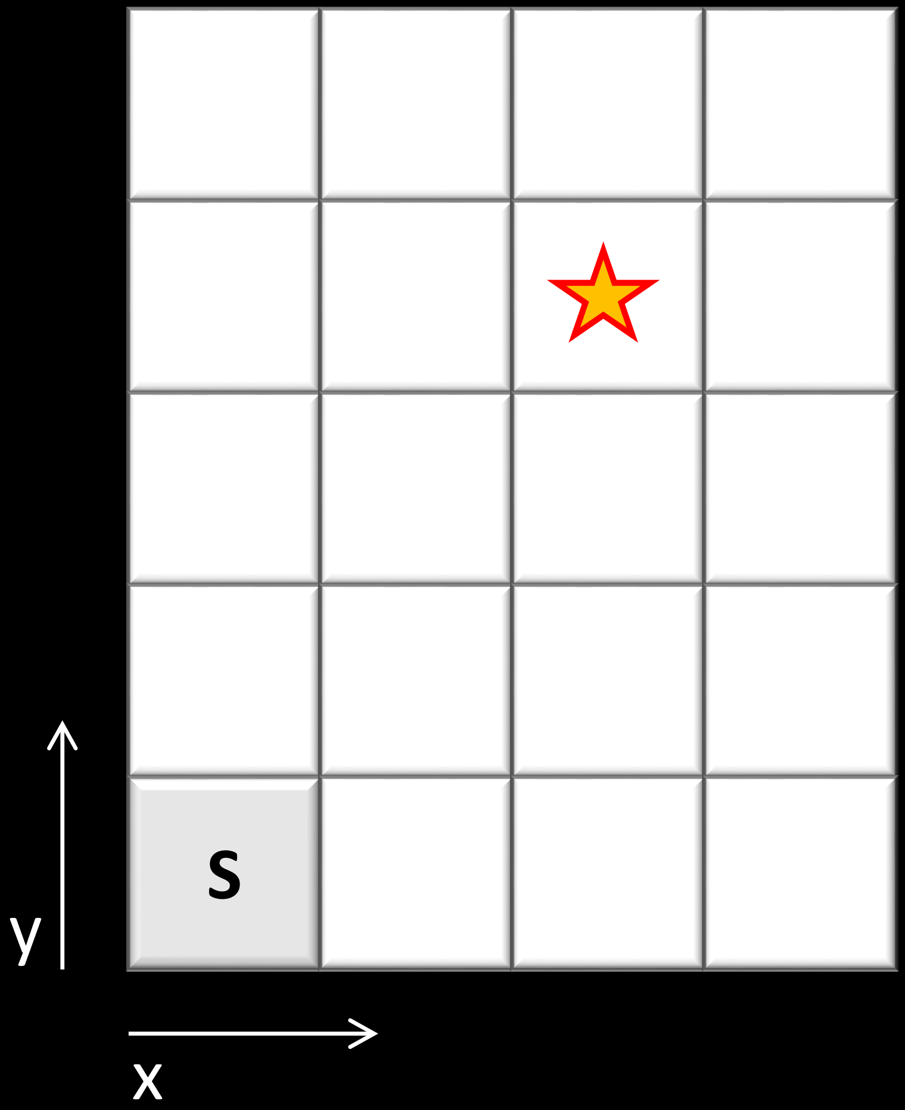
Real world → Configuration Space → Map Representation → Graph Construction → Graph Search

- Depth first search
- Breadth first search
- Lowest-Cost first search
- Greedy search
- A* search

Search Algorithms, General

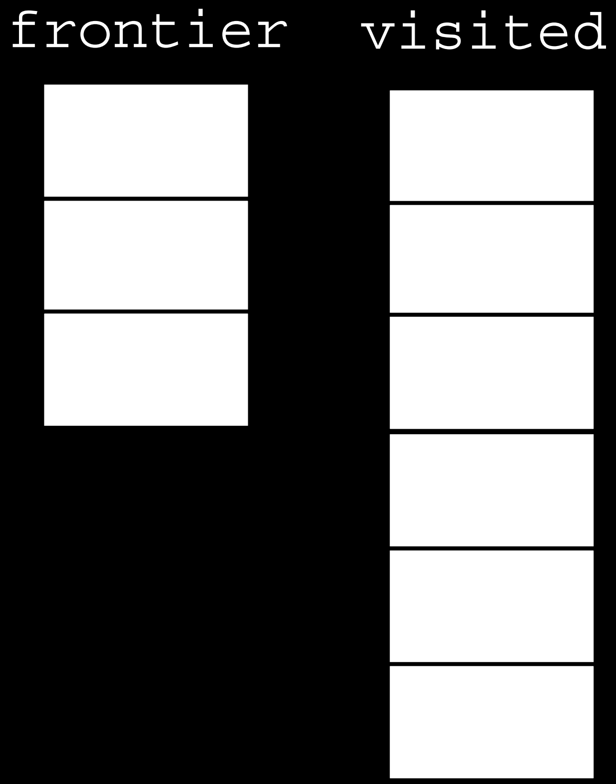
- Common graph structure
  - For every node, n
  - you have a set of actions, a
  - that moves you to a new node, n’

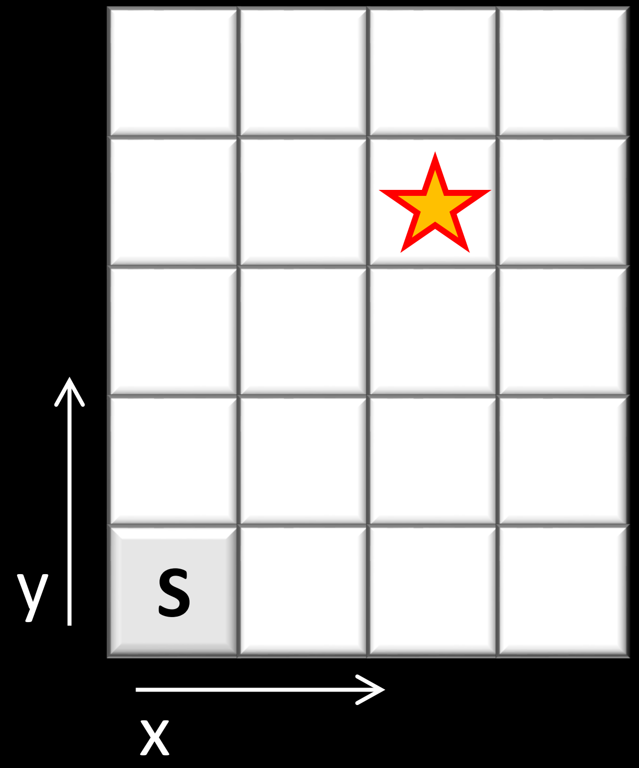




Search Algorithms, General

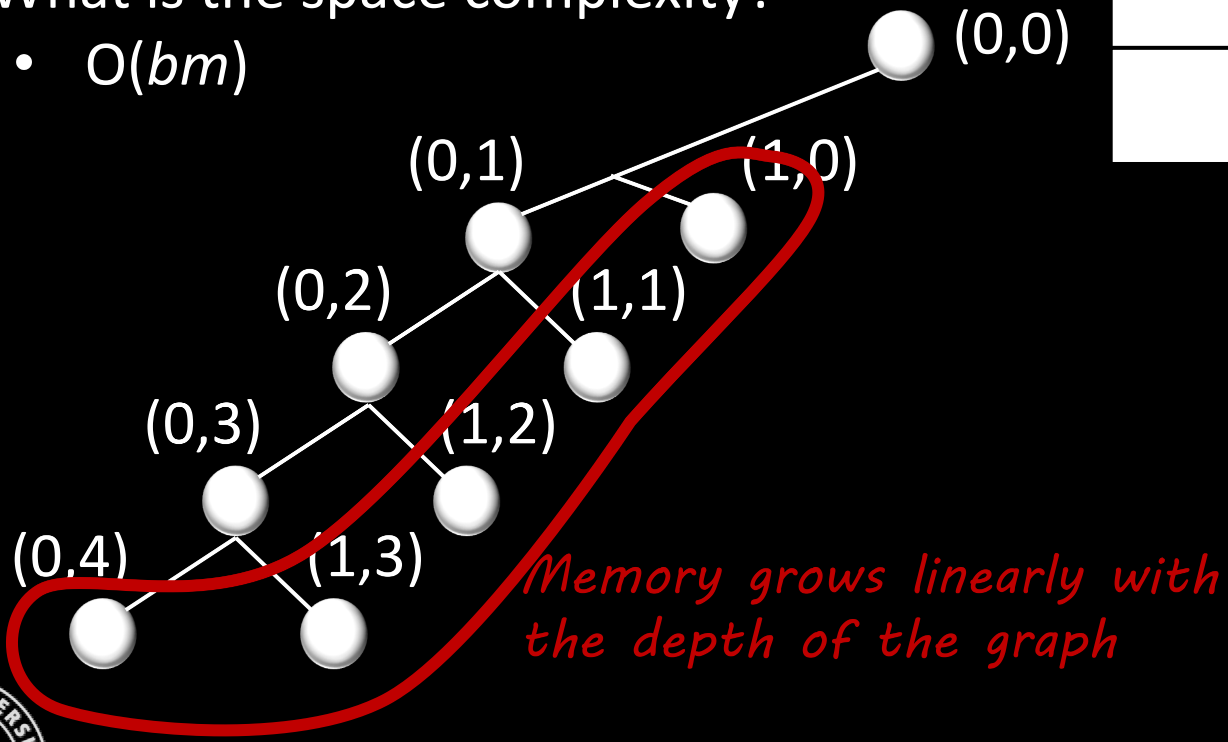
```
n = state(init)
frontier.append(n)
while(frontier not empty)
   n = pull state from frontier
   append n to visited
   if n = goal, return solution
   for all actions in n
      n' = a(n)
      if n' not visited
           append n' to frontier
```



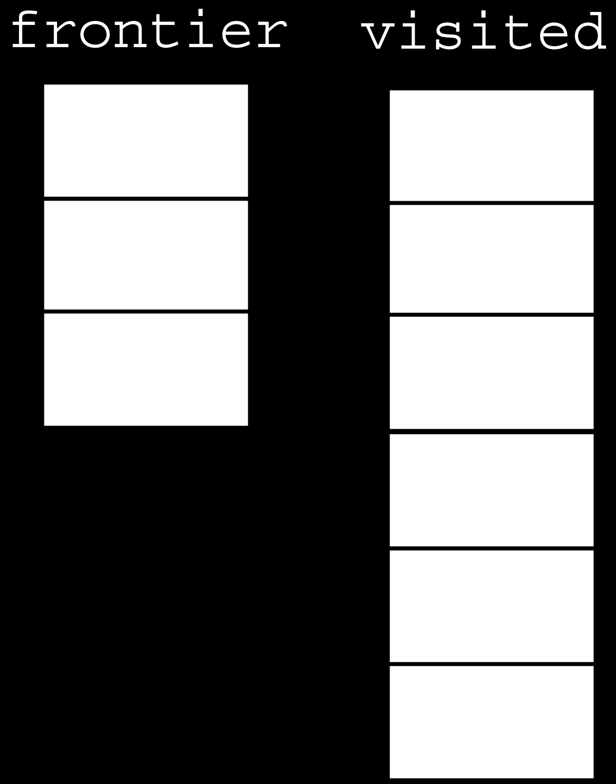


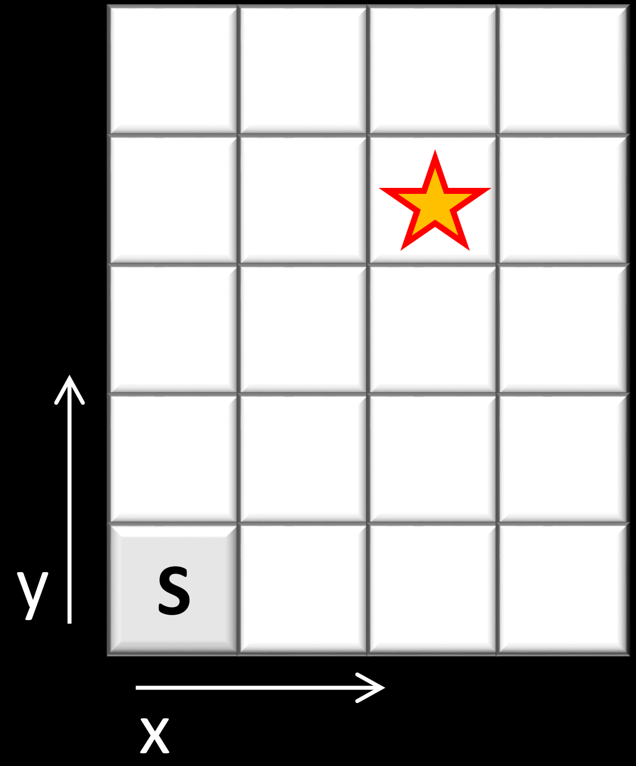
Depth First Search (DFS)

- Is it complete?
  - Yes, but only on finite graphs
- What is the time complexity?
  - $O(b^m)$
- What is the space complexity?
  - $O(bm)$



Last-In First-Out (LIFO) Buffer

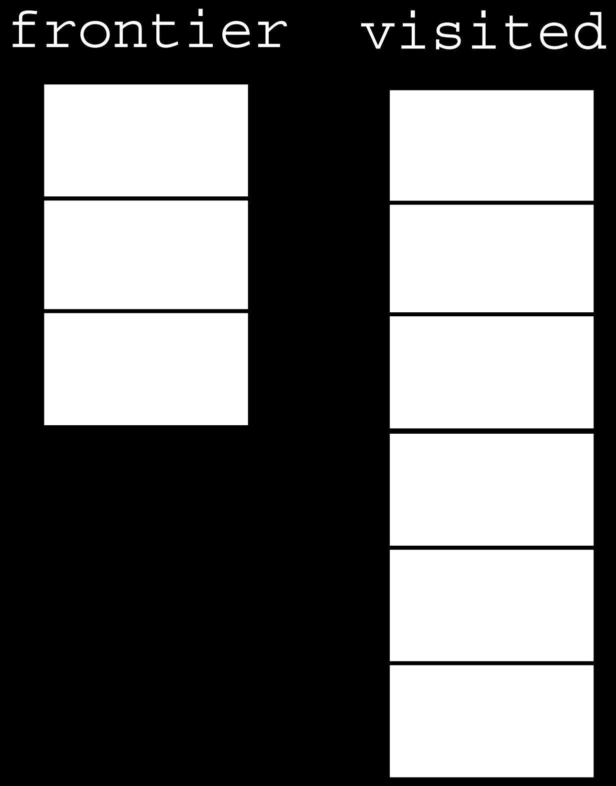


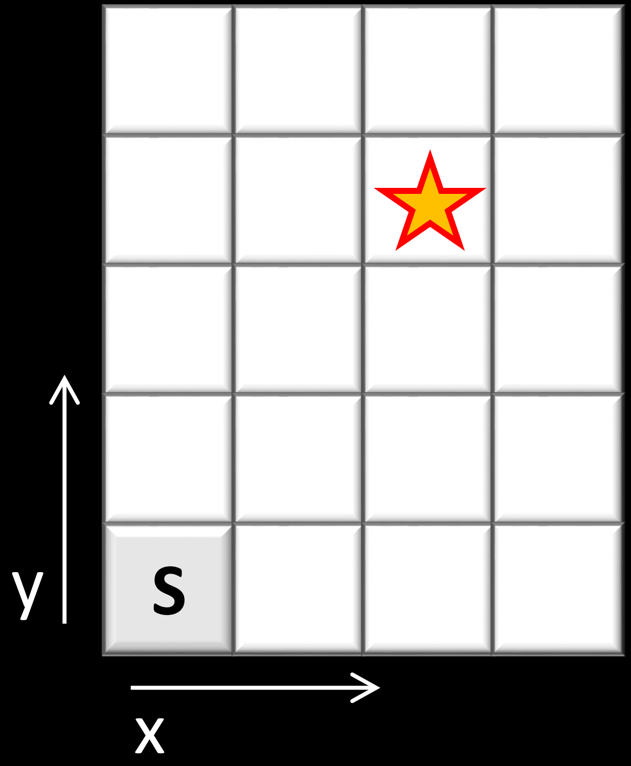


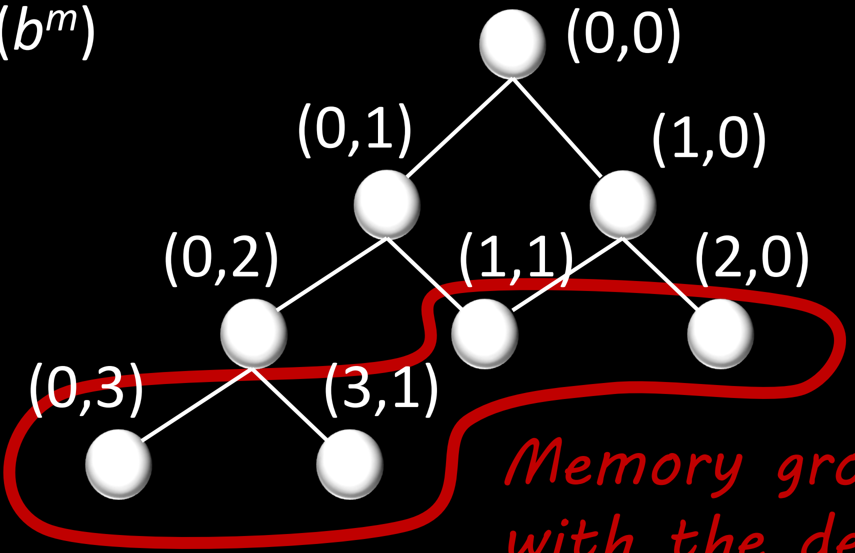
Breadth First Search (BFS)

- Is it complete?
  - Yes, as long as b is finite
- Is it optimal?
  - Yes
- What is the time complexity?
  - $O(b^m)$
- What is the space complexity?
  - $O(b^m)$

First-In First-Out (LIFO) Buffer



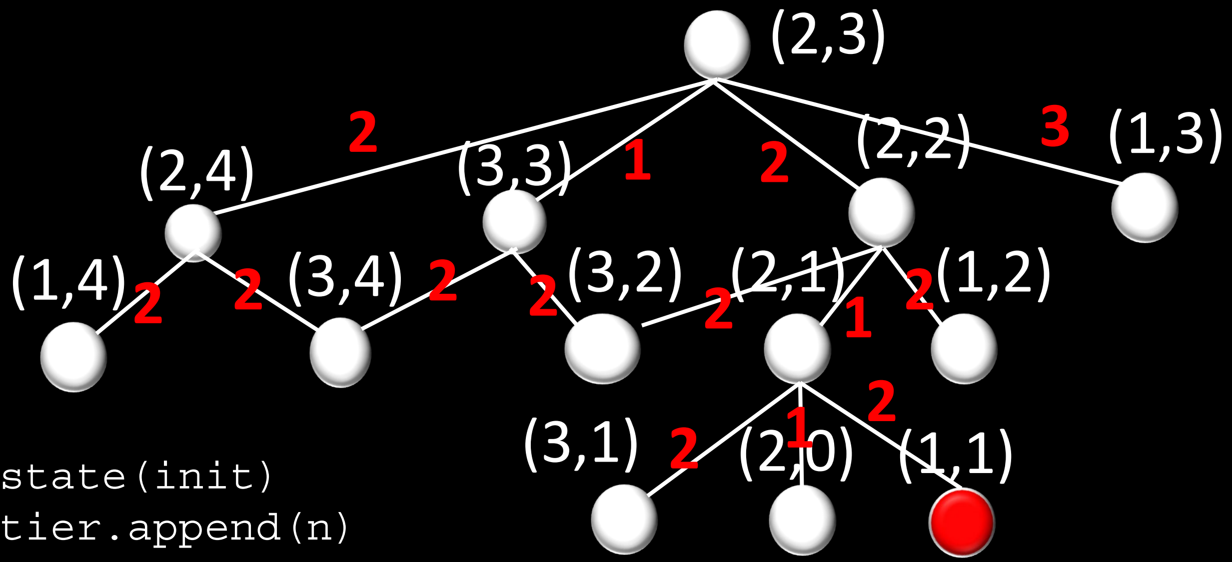




Memory grows exponentially with the depth of the graph

Lowest-Cost First Search

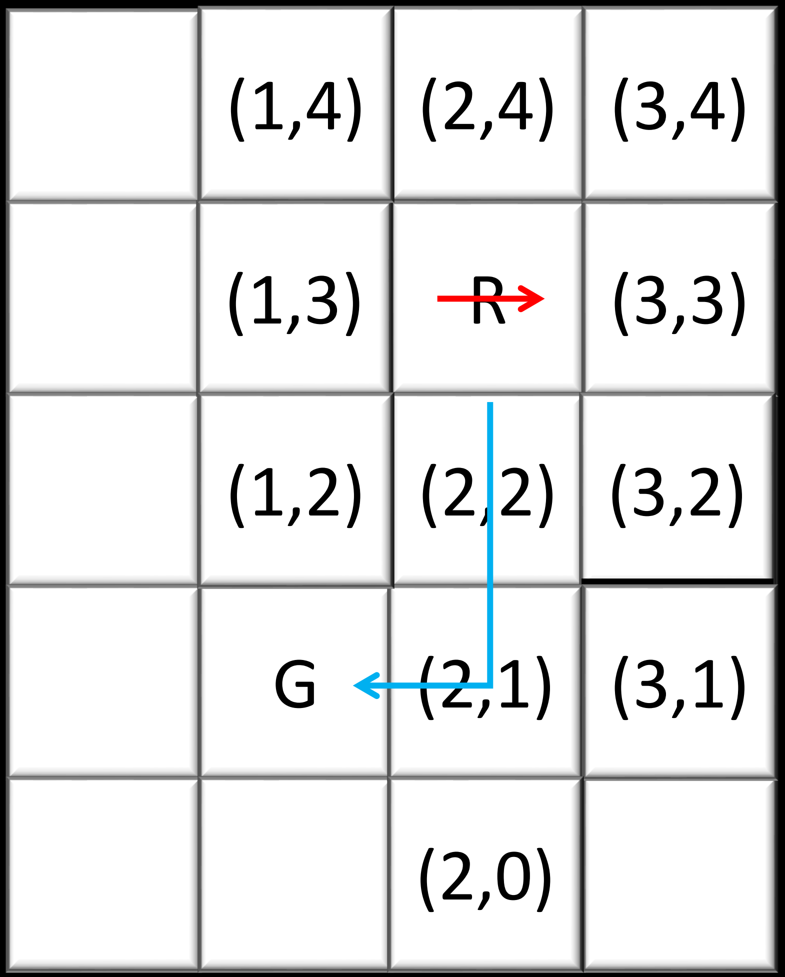
- Consider parent cost!




```
n = state(init)
frontier.append(n)
while(frontier not empty)
    n = pull state from frontier
    visited.append(n)
    if n = goal, return solution
    for all actions in n
        n' = a(n)
        if n' not visited
            priority = heuristic(goal,n')
            frontier.append(priority)
```

Cost

- Go straight, cost 1
- Turn one quadrant, cost 1



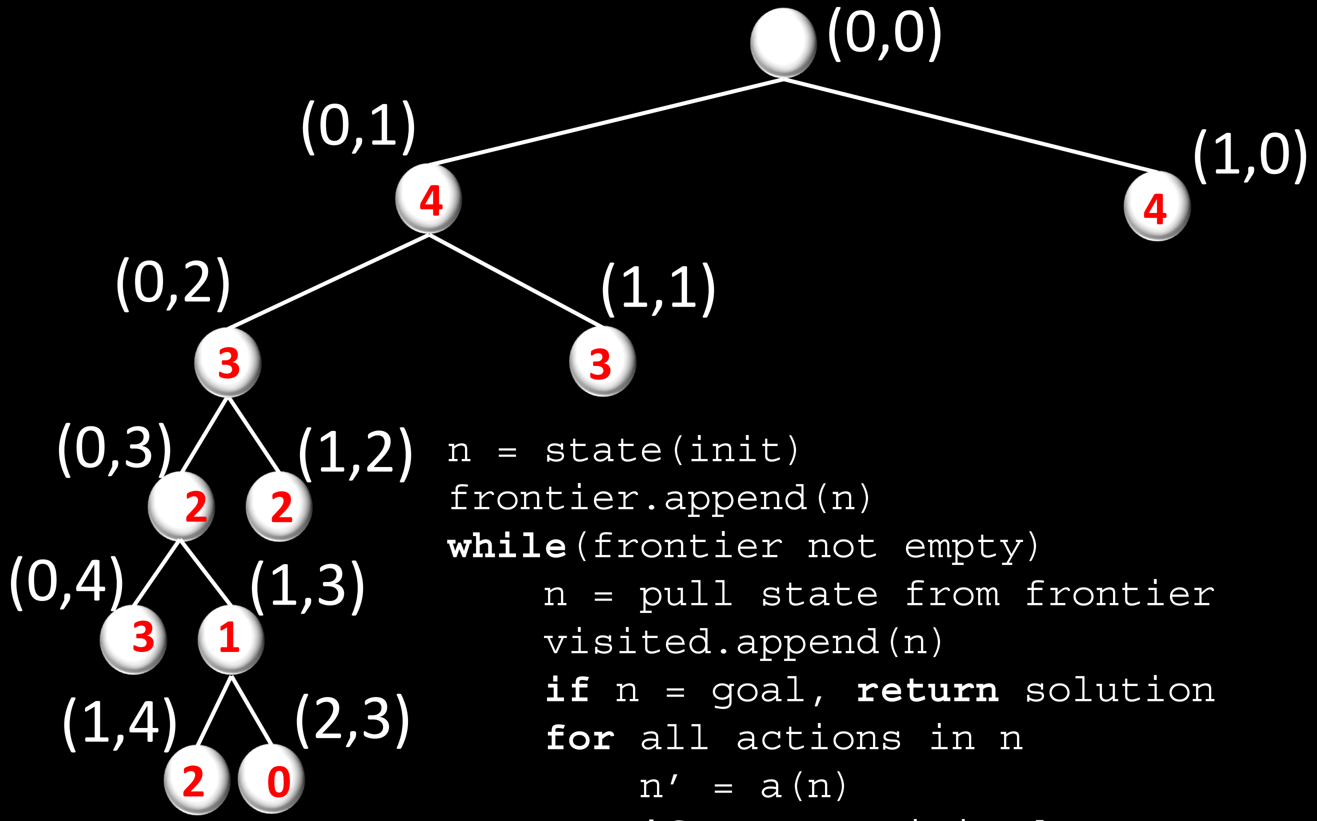
Data structure

- n.state
- n.parent
- n.cost
- n.action

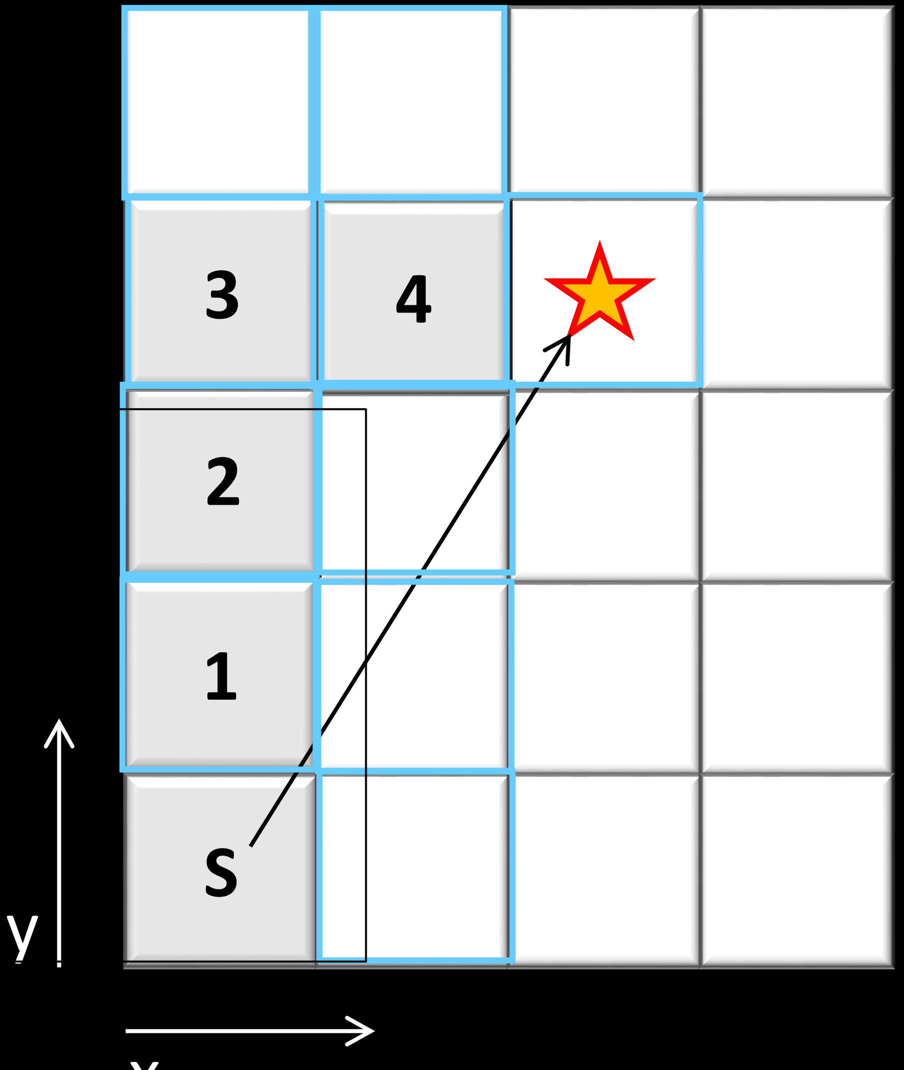
Informed Search

Search order: N, E, S, W

- Greedy Search
  - Define a heuristic to target the goal



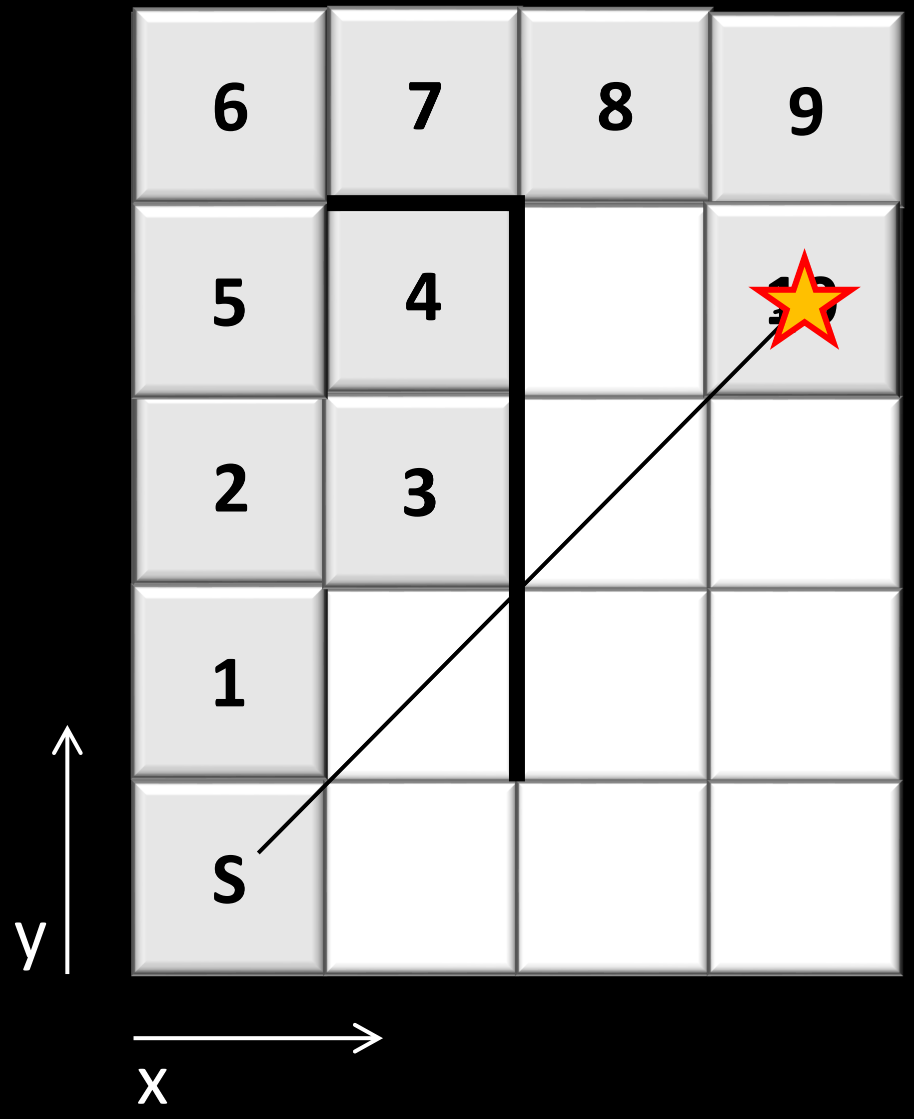

```
n = state(init)
frontier.append(n)
while(frontier not empty)
    n = pull state from frontier
    visited.append(n)
    if n = goal, return solution
    for all actions in n
        n' = a(n)
        if n' not visited
            priority = heuristic(goal,n')
            frontier.append(priority)
```



Informed Search

Search order: N, E, S, W

- Greedy Search
  - Complete?
    - No
  - Time complexity?
    - $O(b^m)$
  - Space complexity?
    - $O(b^m)$
  - Optimal?
    - no…



Search Algorithms, General

- Breadth First Search
  - *Complete* and optimal
  - …but searches *everything*
- Lowest-Cost First Algorithm *Considers parent cost*
  - *Complete and optimal*
  - …but it wastes time exploring in directions that aren't promising
- Greedy Search *Considers goal*
  - *Complete (in most cases)*
  - …only explores promising directions

A*

Can we do better?

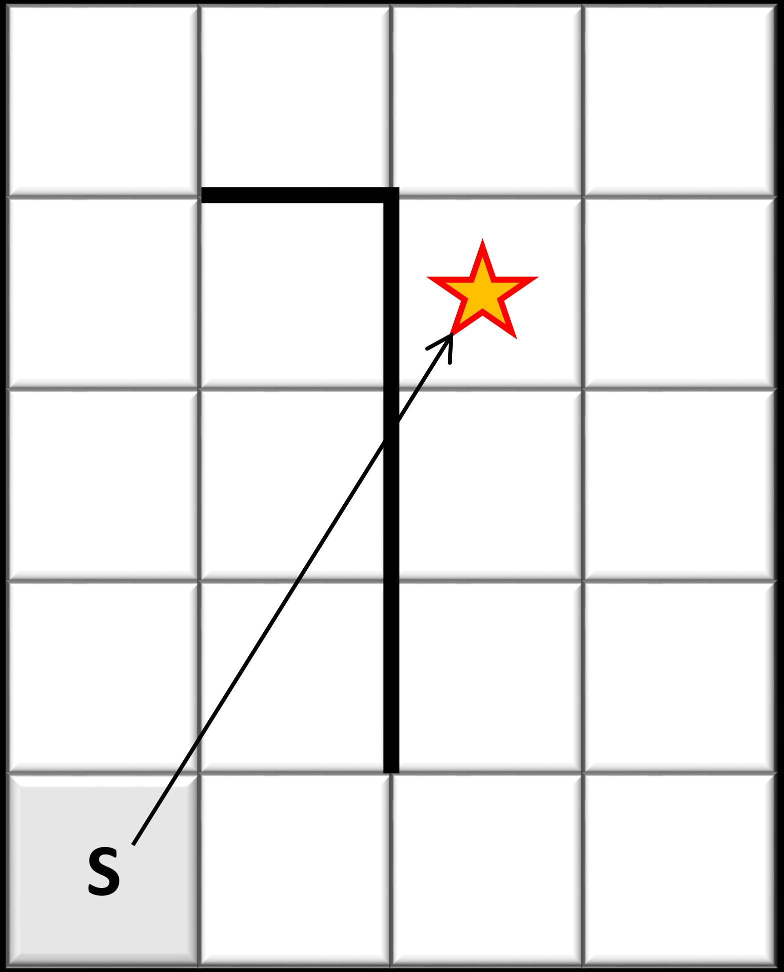
Informed Search

Search order: N, E, S, W

- A* ("A-star")

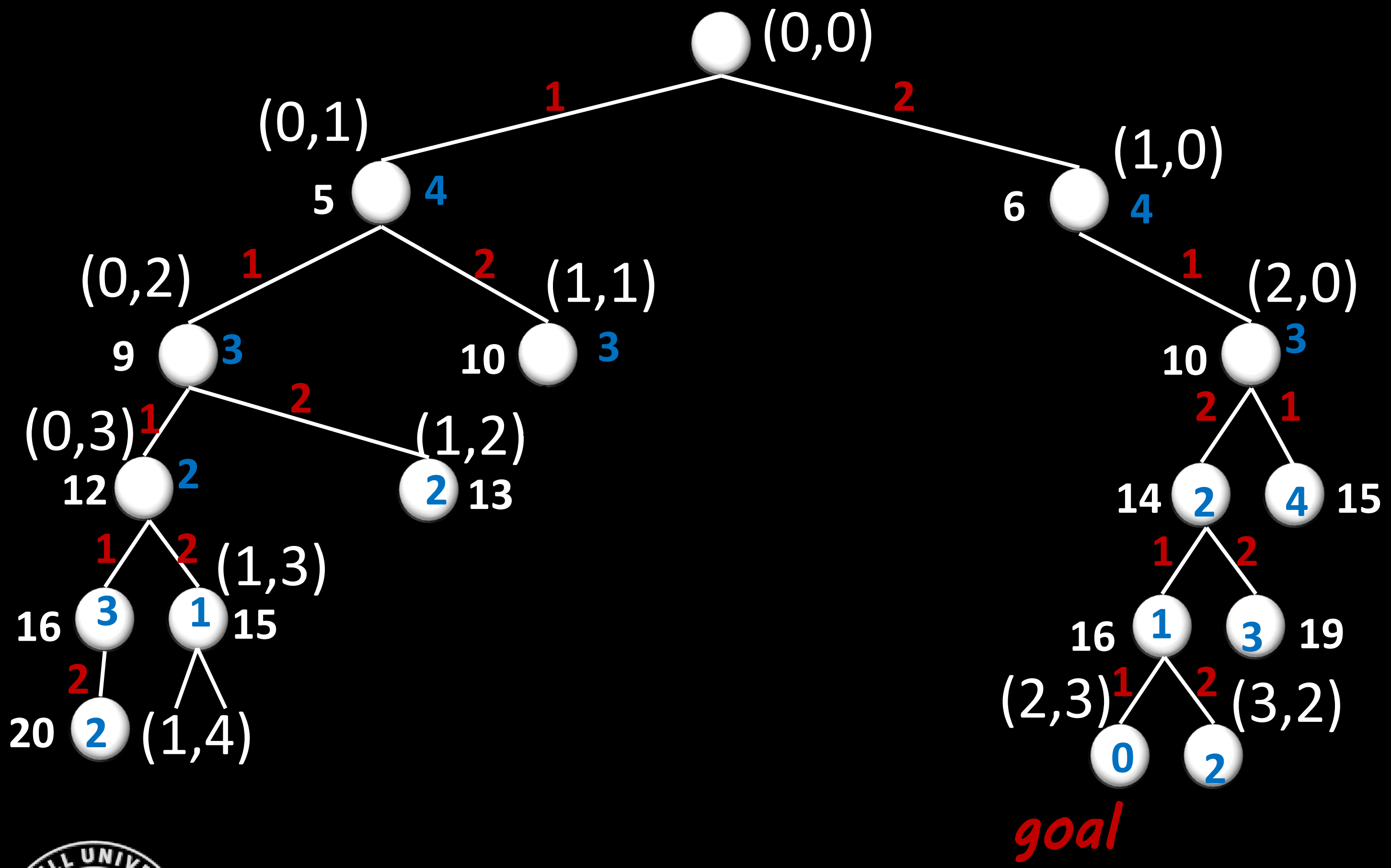
```
n = state(init)
frontier.append(n)
while(frontier not empty)
    n = pull state from frontier
    if n = goal, return solution
    for all actions in n
        n' = a(n)
        if (n' not visited)
              priority = heuristic(goal,n')+cost
              frontier.append(priority)
        if (visited and n'.cost < n_old.cost)
             visited.append(n')
```

Find a treasure



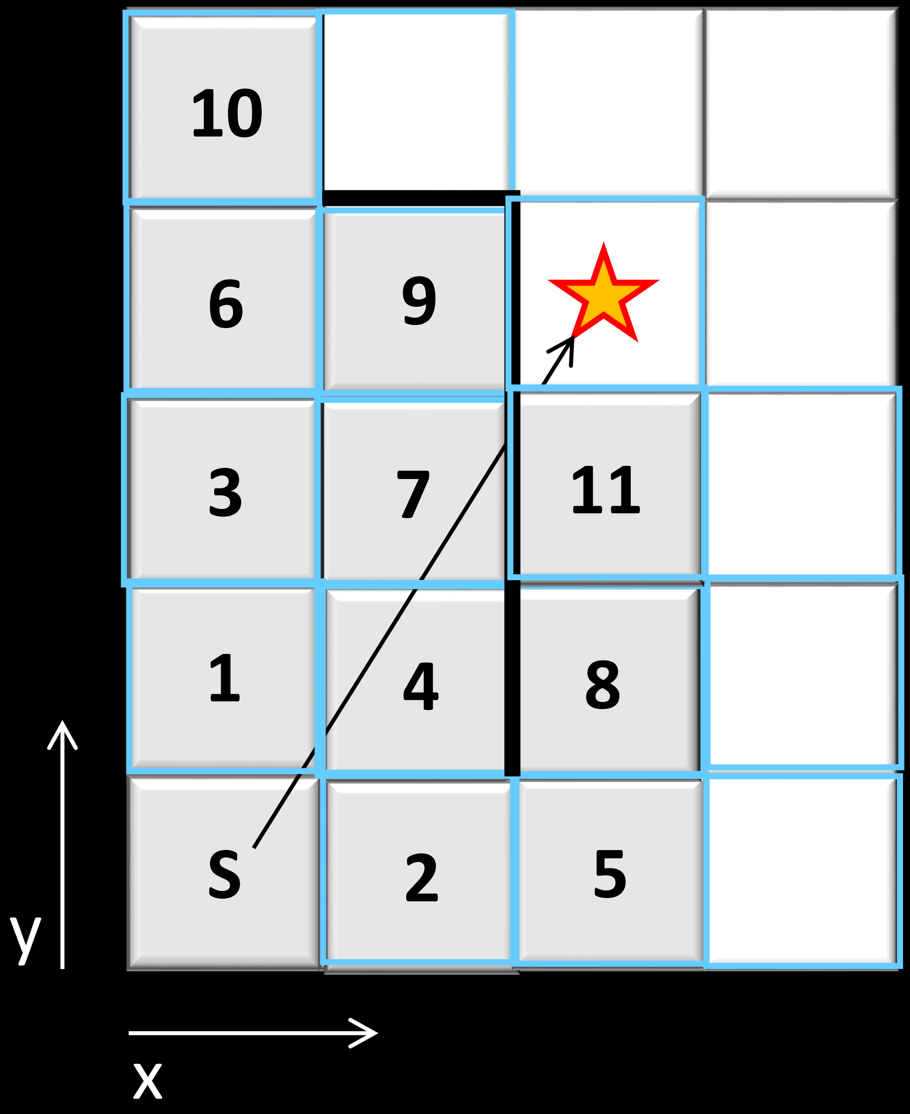
Informed Search

- A* ("A-star")
  - Cost and goal heuristic



Search order: N, E, S, W

Find a goal

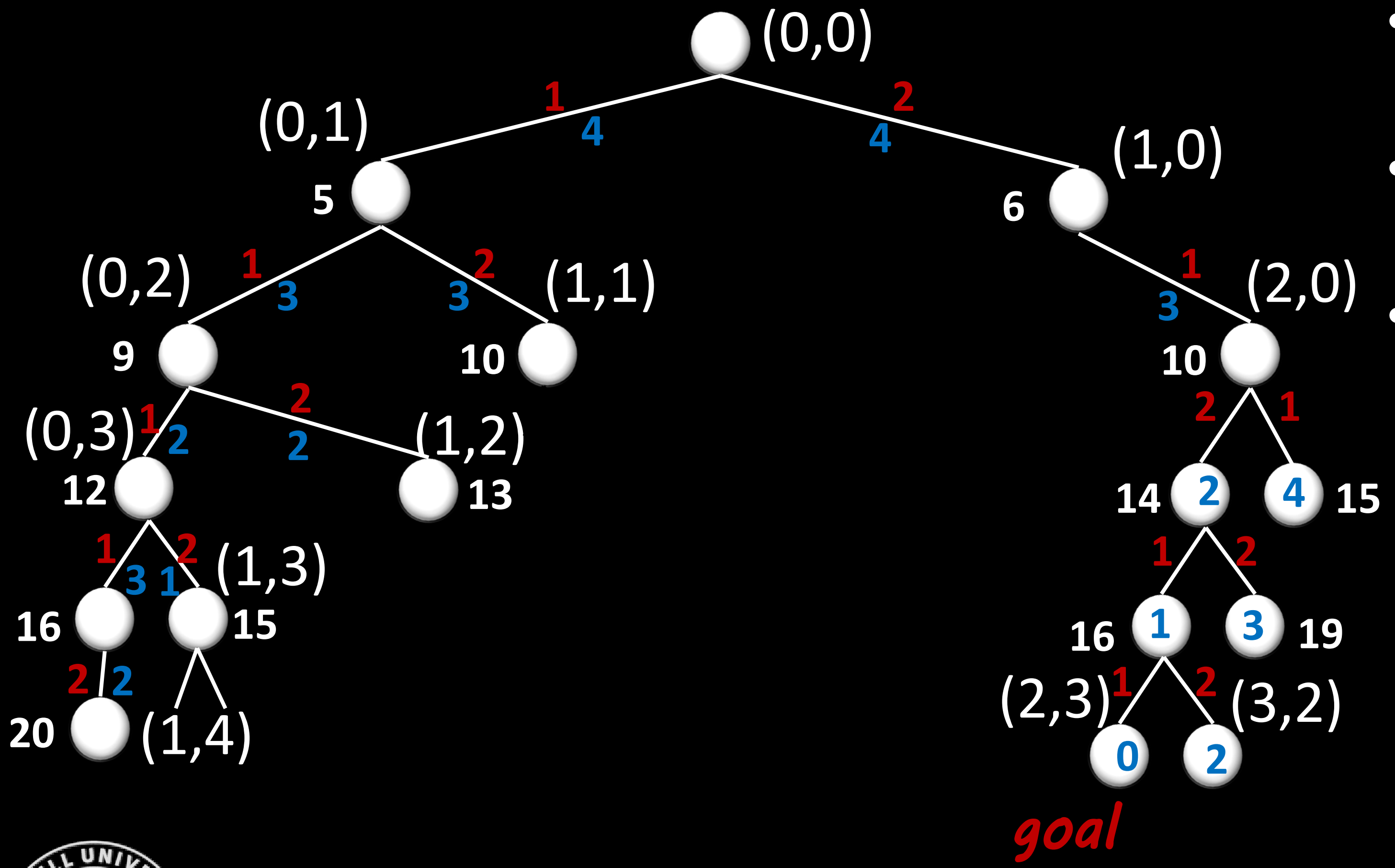


A* Search

- What if the heuristic is too optimistic? *admissible heuristic*
  - Estimated cost < true cost
- What if the heuristic is too pessimistic? *inadmissible heuristic*
  - Estimated cost > true cost
  - No longer guaranteed to be optimal
- What if the heuristic is just right?
  - Pre-compute the cost between all nodes
  - Feasible for you?

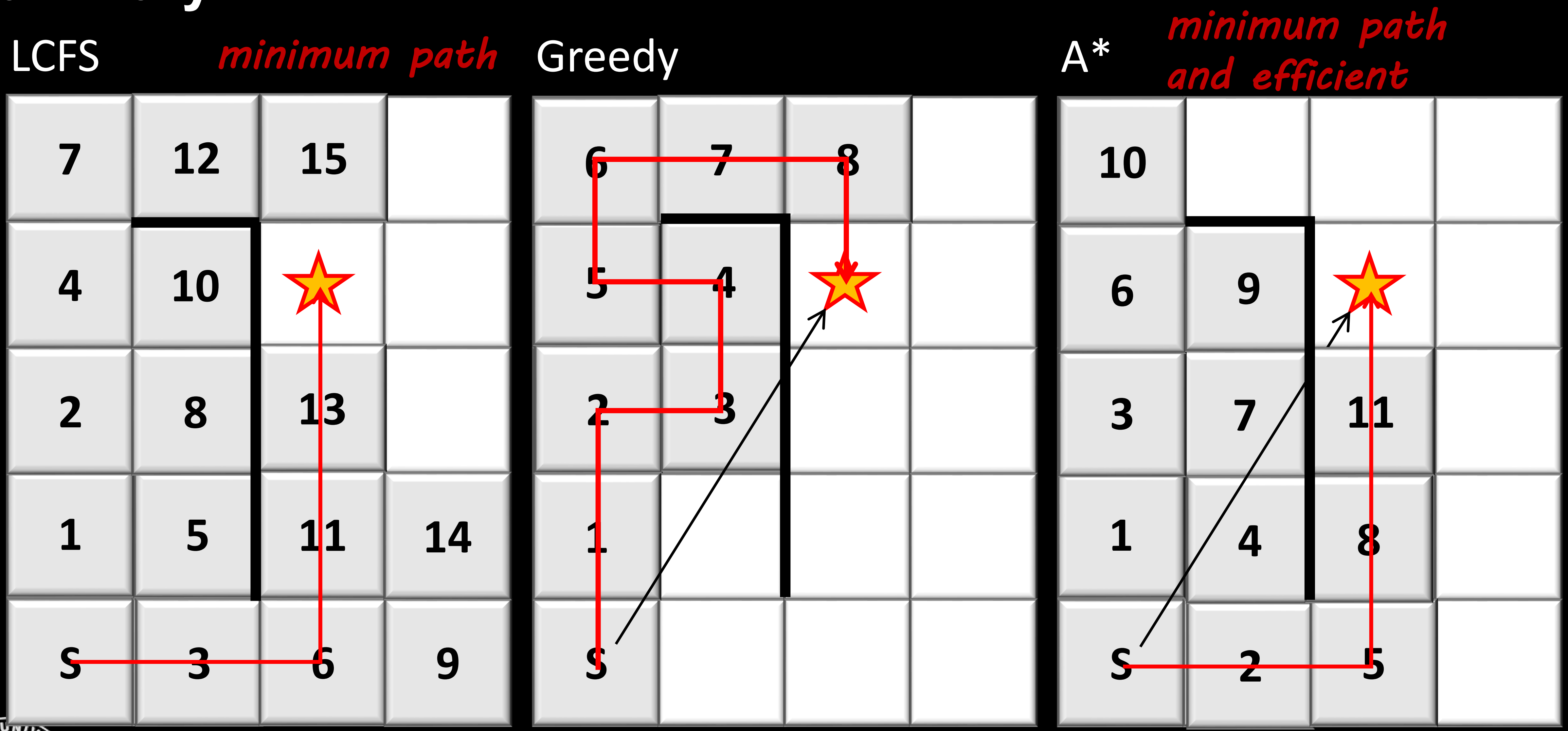
Informed Search

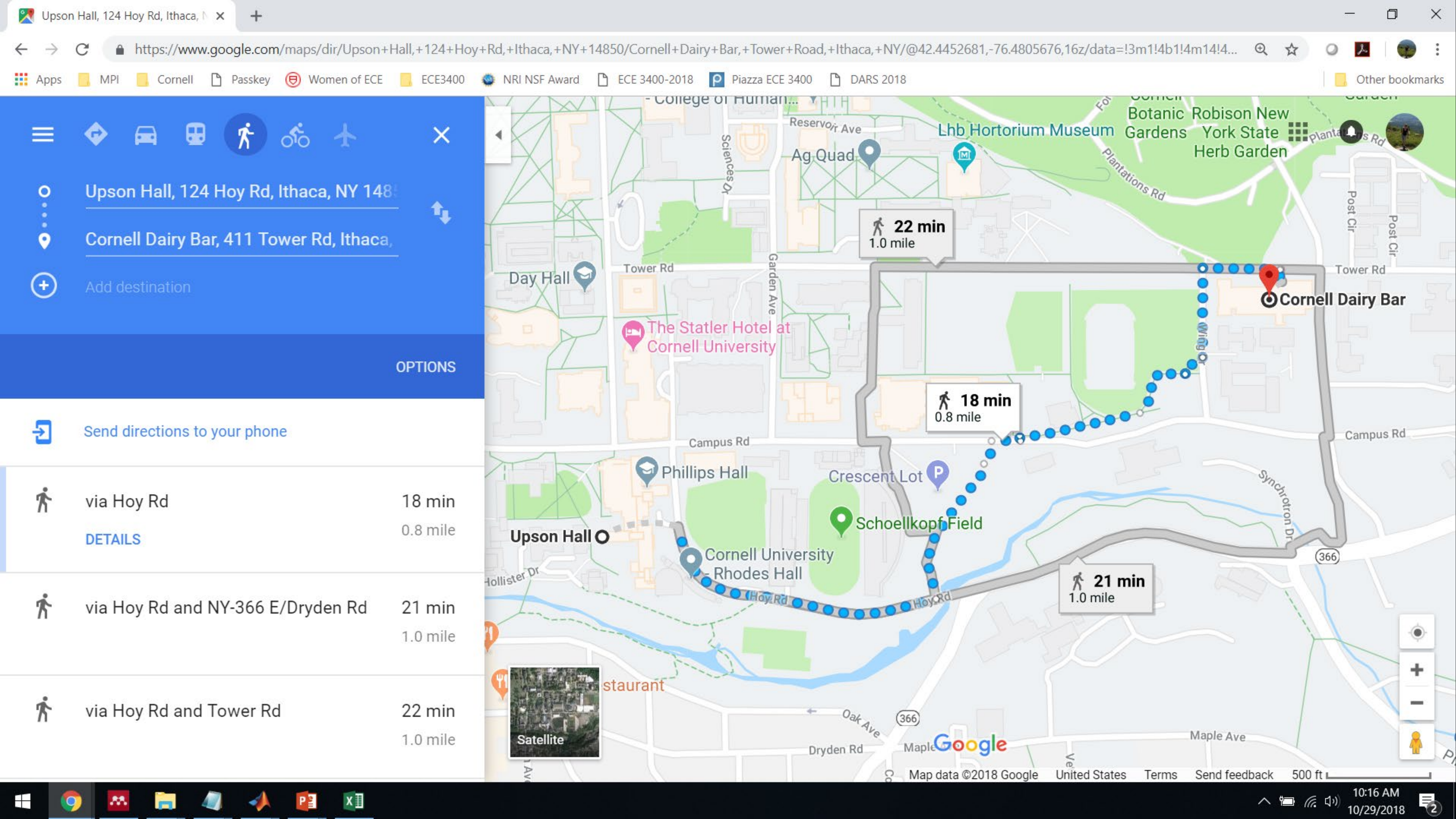
- A* ("A-star")
  - Cost and goal heuristic



- Complete?
  - Yes!
- Time Complexity
  - $O(b^m)$
- Space Complexity
  - $O(b^m)$
- Optimal?
  - Yes, if the heuristic is admissible!

Summary





Upson Hall, 124 Hoy Rd, Ithaca, N
https://www.google.com/maps/dir/Upson+Hall,+124+Hoy+Rd,+Ithaca,+NY+14850/Cornell+Dairy+Bar,+Tower+Road,+Ithaca,+NY/@42.4452681,-76.4805676,16z/data=!3m1!4b1!4m14!4...
Apps
MPI
Cornell
Passkey
Women of ECE
ECE3400
NRI NSF Award
ECE 3400-2018
Piazza ECE 3400
DARS 2018
Other bookmarks
Upson Hall, 124 Hoy Rd, Ithaca, NY 148
Cornell Dairy Bar, 411 Tower Rd, Ithaca,
Add destination
OPTIONS
Send directions to your phone
via Hoy Rd
18 min
0.8 mile
DETAILS
via Hoy Rd and NY-366 E/Dryden Rd
21 min
1.0 mile
via Hoy Rd and Tower Rd
22 min
1.0 mile
22 min
1.0 mile
18 min
0.8 mile
21 min
1.0 mile
Upson Hall
Cornell Dairy Bar
Day Hall
The Statler Hotel at Cornell University
Phillips Hall
Crescent Lot
Schoellkopf Field
Cornell University - Rhodes Hall
Ag Quad
Lhb Hortorium Museum
Botanic Gardens
Robison New York State Herb Garden
Tower Rd
Campus Rd
Hoy Rd
Garden Ave
Reservoir Ave
Plantations Rd
Synchrotron Dr
Oak Ave
Dryden Rd
Maple Ave
Hollister Dr
366
Satellite
Map data ©2018 Google
United States
Terms
Send feedback
500 ft
10:16 AM
10/29/2018

Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

ECE 4160/5160
MAE 4910/5910

Fast Robots

Markov Processes
&
Bayes Filter

Bayes Theorem

+

Robot-Environment Model

+

Markov Assumption

=

Bayes Filter

Bayesian Inference

likelihood

prior

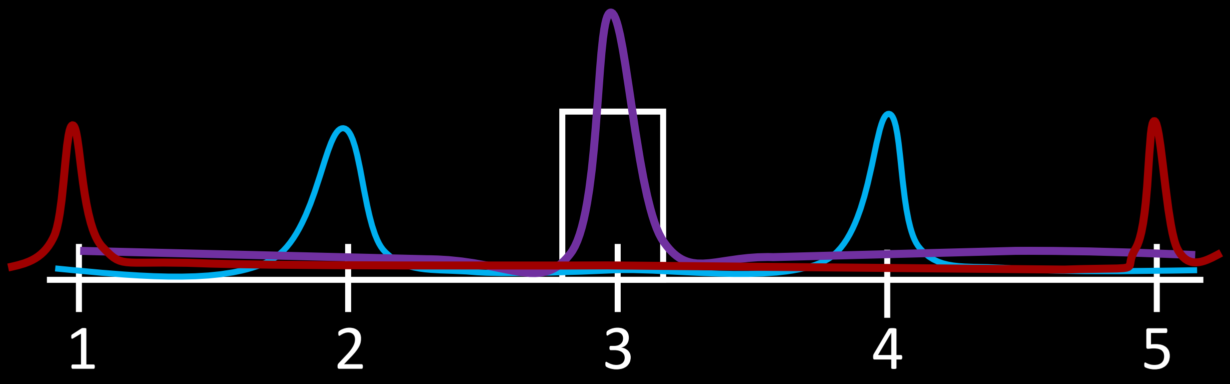
$$P(x|z) = \frac{P(z|x)\,P(x)}{P(z)}$$

posterior

marginal likelihood (constant)

- z = Sensor data
- x = Robot state/ location

- Lost robot example
  - Sensor measures distance to the door
  - $p(X_0 = 1 \text{ or } 2 \text{ or } 3 \text{ or } 4 \text{ or } 5) = 1/5$
  - $p(x|z)$ can be hard to compute
  - What is $p(z|x)$?
  - If Z=1, where are you most likely to be?
  - If Z=0, where are you most likely to be?
  - If Z=2, where are you most likely to be?



Bayes Theorem

+

Robot-Environment Model

+

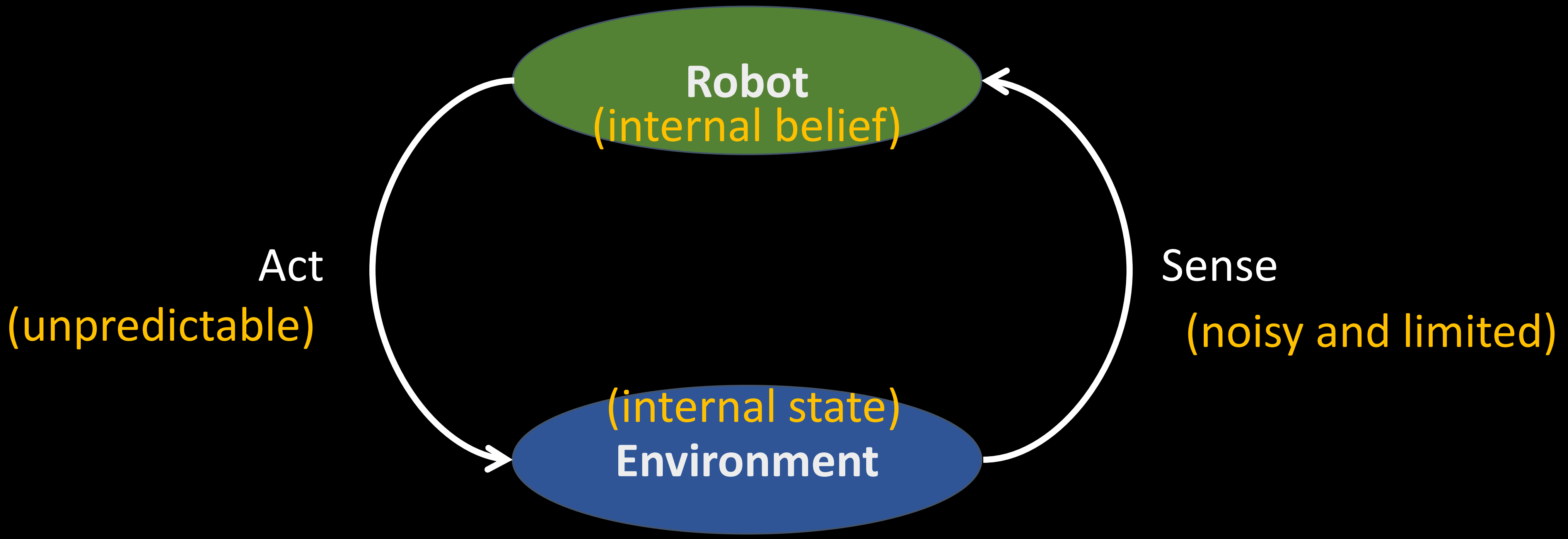
Markov Assumption

=

Bayes Filter

Robot-Environment Interaction

Robot-Environment Interaction



- Two fundamental types of interaction between a robot and its environment:
  - Sensor Measurements/Observations
  - Control Actions

Robot-Environment Model

- Helps us express a robot-environment interaction using probability
  - Typically modeled as a discrete time system
    - The state at time t will be denoted by as x_t
    - A sensor measurement at time t will be denoted as z_t
    - A control action will be denoted by u_t
      - Induces a transition from state x_{t-1} to x_t

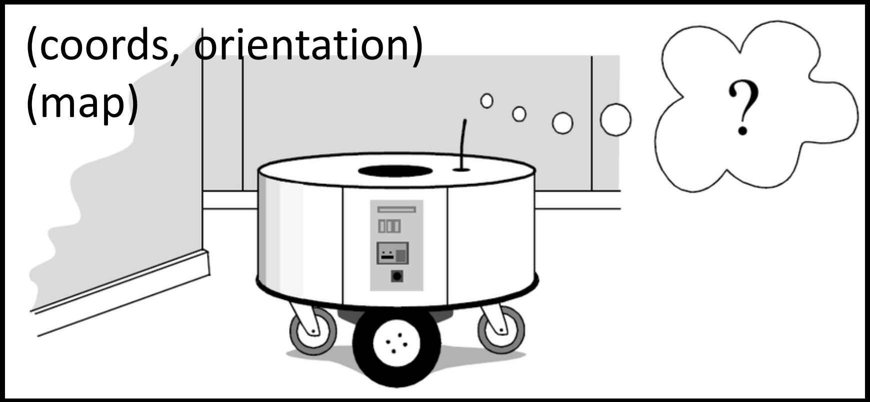
Conventions as per Siegwart, R., Nourbakhsh, I.R. and Scaramuzza, D., 2011. Introduction to autonomous mobile robots. MIT press.

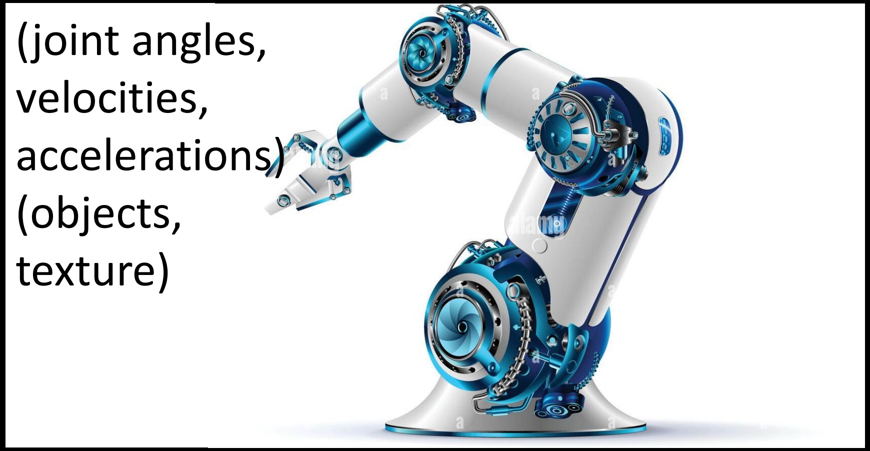
Robot-Environment Model

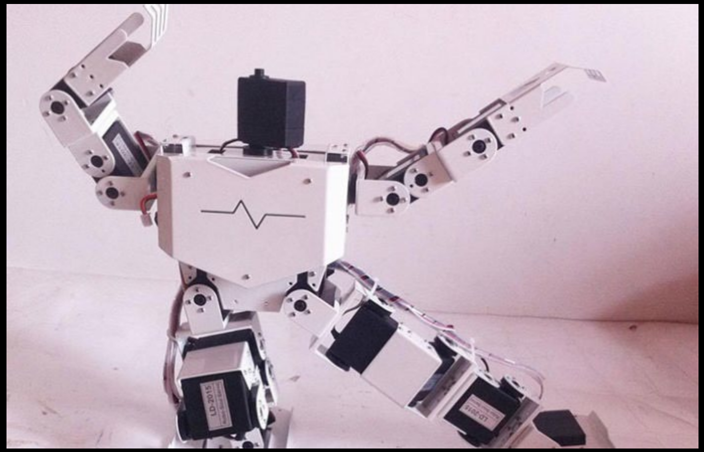
- (Arbitrary) Assumptions
  - The robot executes a control action u_t first and then takes a measurement z_t
  - There is one control action per time step t
    - Control actions include a legal action "*do-nothing*"
  - There is only one measurement z per time step t
  - Shorthand Notation: $x_{t1:t2} = x_{t1}, x_{t1+1}, x_{t1+2}, \ldots, x_{t2}$

Robot State

- The state, x, includes:
  - Robot Specific:
    - Pose, Velocity, Sensor status, etc.
  - Environment Specific:
    - Static variables
      - location of walls
    - Dynamic variables
      - Whereabouts of people in the vicinity of the robot
  - …context-specific







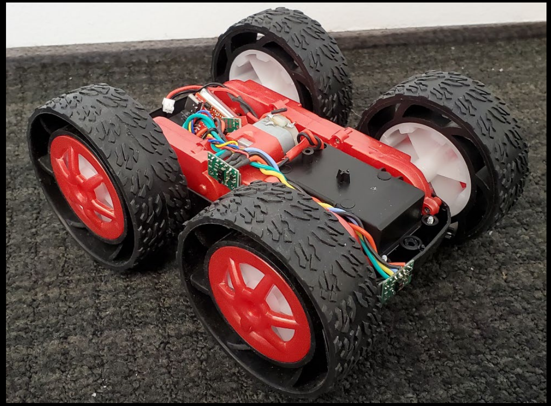
Sensor Measurements/Observations

- z_t
  - Tend to increase the robot's knowledge

balance

Control Actions

- u_t
  - …change the state of the world
  - carry information about the change of the robot state in the time interval (t-1:t]
  - Tends to induce loss of knowledge



Probabilistic Generative Laws

- The evolution of state and measurements is governed by probabilistic laws
  - State: How is x_t generated stochastically?
  - Measurements: How is z_t generated stochastically?

State Generation

- x_t depends on $x_{0:t-1}$, $z_{1:t-1}$ and $u_{1:t}$

$$p(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t})$$

…intractable!

Bayes Theorem

+

Robot-Environment Model

+

Markov Assumption

=

Bayes Filter

Markov Assumption

Markov Assumption

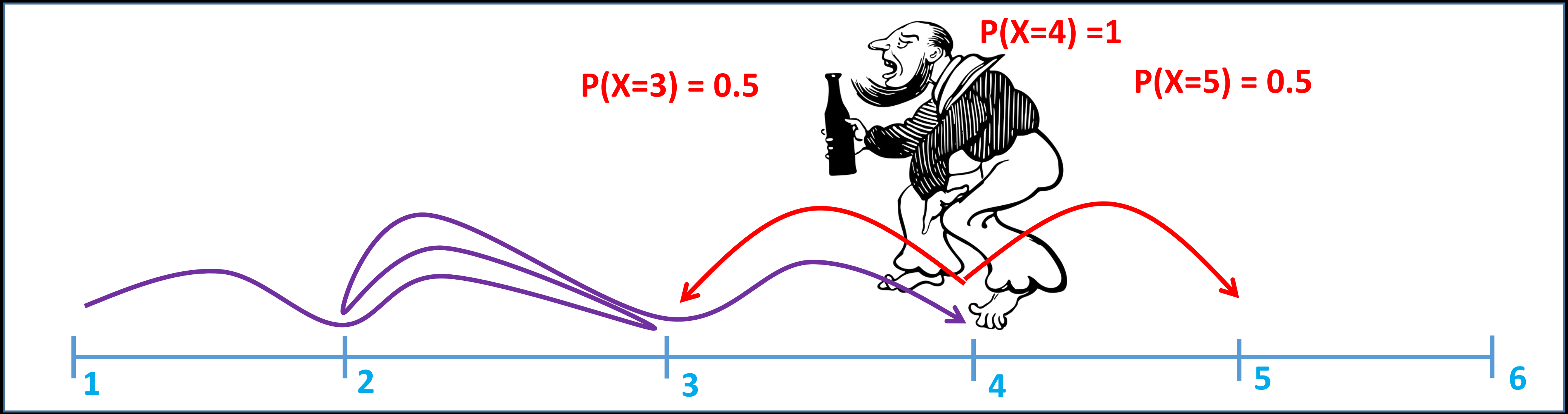
The Markov assumption postulates that past and future data are independent if one knows the current state

- A stochastic model/process that obeys the Markov assumption is a Markov model
- (This does not mean that x_t is deterministic based on x_{t-1})
- *If* we can model our robot as a Markov process… ???
  - We can recursively estimate x_t using
    - x_{t-1}, z_t, u_t
    - *But not* $x_{0:t-1}, z_{1:t-1}, u_{1:t}$*!*
    - Tractable!

Andrey Markov (1856–1922) was a Russian mathematician best known for his work on stochastic processes

Drunkard's walk!

- Random walk on the number line
  - At each step, the position may change by +1 or −1 with equal probability
- The transition probabilities depend only on the current position, not on the manner in which the position was reached
- This is a Markov Process!



Coin Purse

- Contents
  - 5 quarters (25¢)
  - 5 dimes (10¢)
  - 5 nickels (5¢)
- Draw coins randomly, one at a time and place them on a table
- Example:
  - X_n = total value of coins on the table after n draws
  - The sequence $\{ X_n : n \in \mathbb{N} \}$ is a stochastic process

- First, I draw a nickel
- What is $X_1 =$ 5¢
- Next, I draw a dime
- What is $X_2 =$ 15¢

$$X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow$$

Coin Purse

- Contents
  - 5 quarters (25¢)
  - 5 dimes (10¢)
  - 5 nickels (5¢)
- Draw coins randomly, one at a time and place them on a table
- Example:
  - X_n = total value of coins on the table after n draws
  - The sequence $\{ X_n : n \in \mathbb{N} \}$ is a stochastic process

- Suppose…
  - In the first six draws, you pick all 5 nickels and 1 quarter
  - $X_6 = 50¢$
- What can we say about X_7?
  - $P(X_7 \geq 0.55) = 1$
- Can you do better?
  - Can you draw a nickel in the 7th draw?
  - $P(X_7 \geq 0.6) = 1$

- Exercise
  - Is this a Markov Model?
  - If not, can you tweak the definition of X_n to make it one?

Coin Purse

- Contents
  - 5 quarters (25¢)
  - 5 dimes (10¢)
  - 5 nickels (5¢)
- Draw coins randomly, one at a time and place them on a table
- Example:
  - X_n = total value of coins on the table after n draws
  - The sequence $\{ X_n : n \epsilon \mathbb{N} \}$ is a stochastic process

- Markov model
  - X_n = {number of quarters, number of dimes, number of nickels} drawn
  - First you pick a nickel
    - $X_1 = \{0,0,1\}$
    - $X_6 = \{1,0,5\}$
  - Now, what can you say about X_7?
    - $p(X_7 \geq 0.6) = 1$
- State space: 6*6*6 = 216 possible states
- …but independent of the number of draws

Robot-Environment Model

State Generative Model

- x_t is generated stochastically from the state x_{t-1}
- x_t depends on $x_{0:t-1}$, $z_{1:t-1}$ and $u_{1:t}$

$$p(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t-1})$$

- If state x_t is modeled under the **Markov Assumption**, then

$$p(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t-1}) = p(x_t \mid x_{t-1}, u_t)$$

  - Knowledge of only the previous state x_{t-1} and control u_t is sufficient to predict x_t

Measurement Generative Model

- Similarly, the process by which measurements are generated are of importance

$$p(z_t|x_{0:t}, z_{1:t-1}, u_{1:t})$$

- If x_t conforms to the **Markov Assumption**, then

$$p(z_t|x_{0:t}, z_{1:t-1}, u_{1:t}) = p(z_t|x_t)$$

  - The state x_t is sufficient to predict the (potentially noisy) measurements
  - Knowledge of any other variable, such as past measurements, controls, or even past states, is irrelevant under the Markov Assumption

Bayes Theorem

+

Robot-Environment Model

+

Markov Assumption

=

Bayes Filter

Robot Belief

- Probabilistic robotics represents beliefs through *posterior conditional probability distributions*
  - probability distributions over state variables conditioned on available data
  - The belief of a robot is the posterior distribution over the state of the environment, given all past sensor measurements and all past controls
    - Belief over a state variable x_t is denoted by $bel(x_t)$:

$$bel(x_t) = p(x_t | z_{1:t}, u_{1:t})$$

  - The (prior) belief is the belief before incorporating the latest measurement z_t

$$\overline{bel}(x_t) = p(x_t | z_{1:t-1}, u_{1:t})$$

Bayes Filter

- A recursive algorithm that calculates the belief distribution from measurements and control data

1. **Algorithm Bayes_Filter** $(bel(x_{t-1}), u_t\ , z_t)$:
2. for all x_t do
3. $\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t | u_t, x_{t-1})\ bel(x_{t-1})$
4. $bel(x_t)\ =\ \eta\, p(z_t | x_t)\ \overline{bel}(x_t)$
5. endfor
6. return $bel(x_t)$

Bayes Filter

- A recursive algorithm that calculates the belief distribution from measurements and control data

1. **Algorithm Bayes_Filter** $(bel(x_{t-1}), u_t\, , z_t)$:
2. $\quad$ for all x_t do

Transition probability /action model

3. $\quad\quad \overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t|u_t, x_{t-1})\; bel(x_{t-1})$ $\quad$ (Prediction step)
4. $\quad\quad bel(x_t) \;=\; \eta\, p(z_t|x_t)\;\; \overline{bel}(x_t)$
5. $\quad$ endfor
6. return $bel(x_t)$

Bayes Filter

- A recursive algorithm that calculates the belief distribution from measurements and control data

1. **Algorithm Bayes_Filter** $(bel(x_{t-1}), u_t, z_t)$:
2. $\boxed{\text{for all } x_t \text{ do}}$
3. $\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t | u_t, x_{t-1}) \; bel(x_{t-1})$ (Prediction step)
4. $\boxed{bel(x_t)} = \eta \, \boxed{\boxed{p(z_t | x_t)} \; \overline{bel}(x_t)}$ (Update/measurement Step)
5. endfor
6. $\boxed{\text{return } bel(x_t)}$

Transition probability /action model

Measurement Probability / Sensor Model

Kalman Filter Implementation

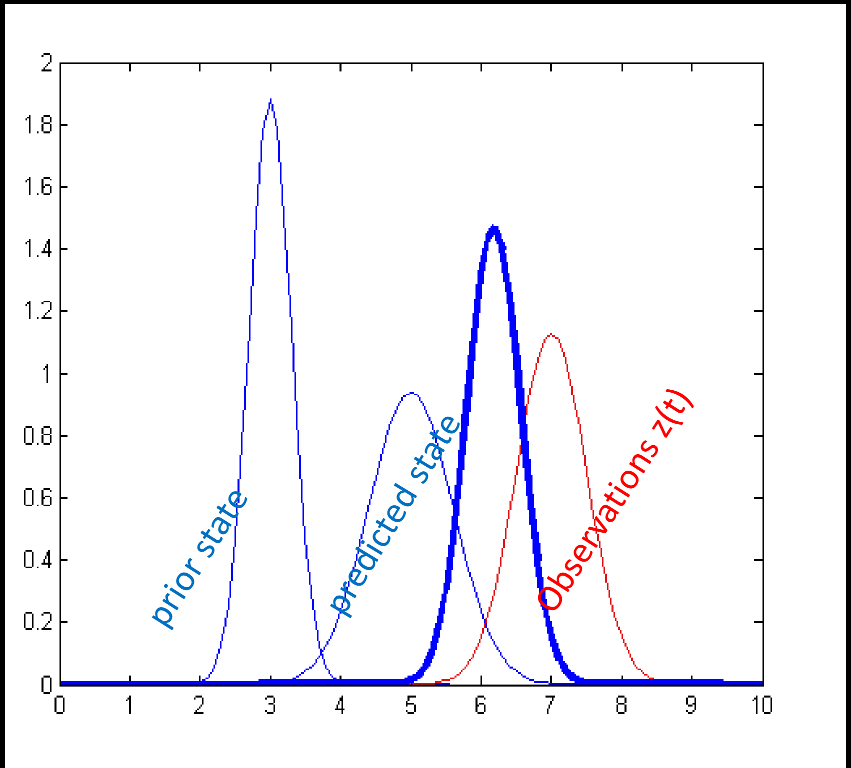
Kalman Filter (μ(t-1), Σ(t-1), u(t), z(t))

1. $\mu_p(t) = A\,\mu(t-1) + B\,u(t)$
2. $\Sigma_p(t) = A\,\Sigma(t-1)\,A^T + \Sigma_u$
3. $K_{KF} = \Sigma_p(t)\,C^T\,(\,C\,\Sigma_p(t)\,C^T + \Sigma_z)^{-1}$
4. $\mu(t) = \mu_p(t) + K_{KF}\,(\,z(t) - C\,\mu_p(t)\,)$
5. $\Sigma(t) = (\,\mathbf{I} - K_{KF}\,C)\,\Sigma_p(t)$
6. Return μ(t) and Σ(t)

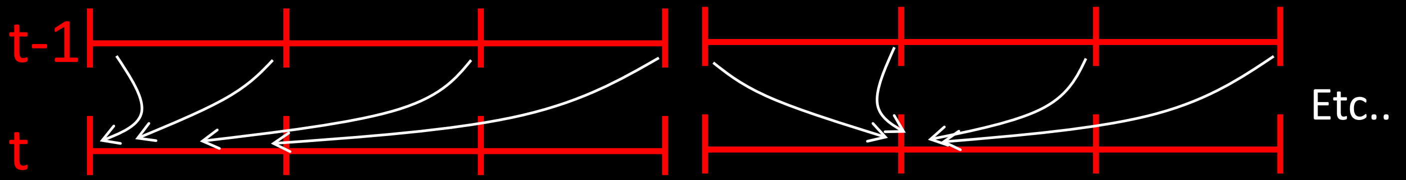
prediction

update

State estimate: $\mu(t)$
State uncertainty: $\Sigma(t)$
Process noise: Σ_u
Kalman filter gain: K_{KF}
Measurement noise: Σ_z



Bayes Filter



1. **Algorithm Bayes_Filter** $(bel(x_{t-1}), u_t\ , z_t)$:
2. for all x_t do

Transition probability /action model

3. $\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t|u_t, x_{t-1})\ bel(x_{t-1})$ (Prediction step)
4. $bel(x_t)\ = \eta\, p(z_t|x_t)\ \overline{bel}(x_t)$ (Update/measurement Step)

Measurement Probability / Sensor Model

5. endfor
6. return $bel(x_t)$

Dynamical Stochastic Model

- $p(x_t | x_{t-1}, u_t)$
  - It is known as the state transition probability
  - It specifies how the robot state evolves over time as a function of robot controls u_t
- $p(z_t | x_t)$
  - It is known as the measurement probability
  - It specifies how the measurements are generated from the robot state x_t
  - Informally, you may think of measurements as noisy projections of the state
- Remember that these predictions are *stochastic and not deterministic*

Bayes Filter - Initial Conditions

- To compute the posterior belief recursively, the algorithm requires an initial belief $bel(x_0)$ at time $t = 0$

Algorithm Bayes_Filter $(bel(x_{t-1}), u_t, z_t)$:

for all x_t do

$\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t | u_t, x_{t-1}) \; bel(x_{t-1})$ (Prediction step)

$bel(x_t) = \eta \, p(z_t | x_t) \; \overline{bel}(x_t)$ (Update/measurement Step)

endfor

return $bel(x_t)$

Bayes Filter - Initial Conditions

- To compute the posterior belief recursively, the algorithm requires an initial belief $bel(x_0)$ at time $t = 0$
- If we know the initial state with absolute certainty, we can initialize a point mass distribution that centers all probability mass on the correct value of x_0 and assign zero everywhere else
- If we are entirely ignorant of the initial state, we can initialize it with a uniform probability distribution over all the possible states

References

1. Thrun, Sebastian, Wolfram Burgard, and Dieter Fox. Probabilistic robotics. MIT press, 2005.
2. http://gki.informatik.uni-freiburg.de/teaching/ws0607/advanced/recordings/BayesFiltering.pdf
3. http://home.deib.polimi.it/restelli/MyWebSite/pdf/E05_h.pdf
4. https://en.wikipedia.org/wiki/Markov_chain
5. https://en.wikipedia.org/wiki/Markov_model
6. Image Sources:
   a. https://blog.robotiq.com/a-brief-history-of-robots-in-manufacturing
   b. https://northwesturbanist.files.wordpress.com/2015/01/20151104_082523.jpg
   c. https://expresswriters.com/is-the-dress-blue-black-white-or-gold-how-it-went-viral/
   d. https://en.wikipedia.org/wiki/Andrey_Markov

Navigation

- Navigation breaks down to: Localization, Map Building, Path Planning