Planning

ME 2984

“Failing to plan is planning to fail.”

— Alan Lakein

SYSTEM REVIEW

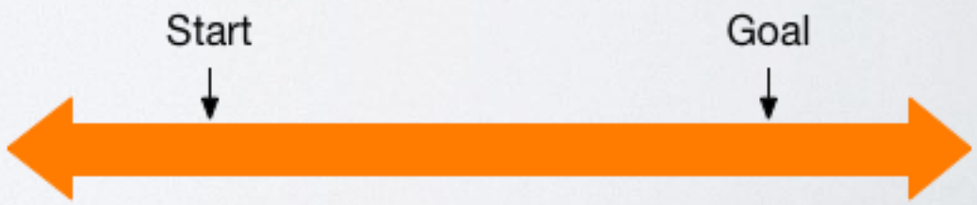
- System Design
- Act
  - Closed-loop Control
    - Kinematics
- Sense
  - Sensing
  - Perception
- Think?

PLANNING

- For some space, find a path that goes from an initial point (start) to and end point (goal)
- Space can be n-dimensional
- Planner is *complete* if it finds a solution in finite time
  - Or returns a failure if no there is no path

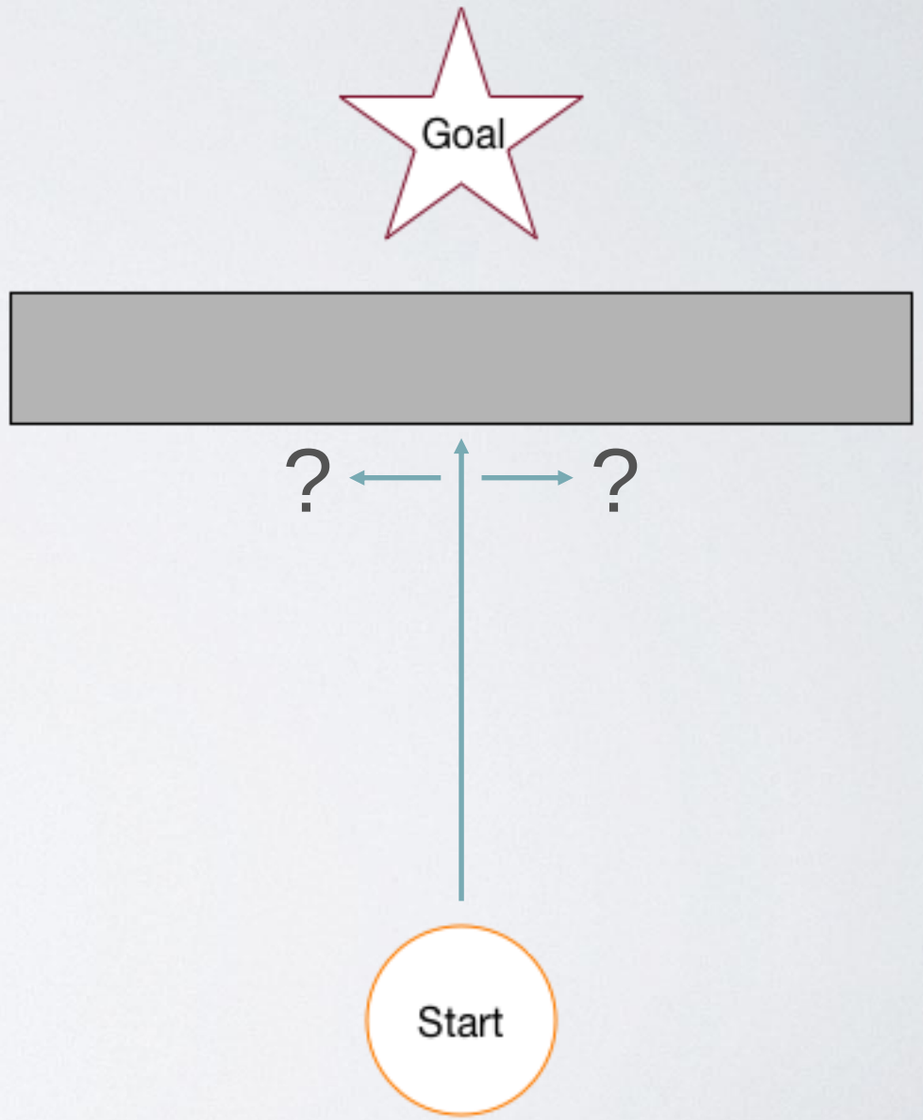
1 DIMENSIONAL SEARCH

- From current position
  - Test if a step in one direction reduces distance to goal
    - If yes, add step to path, step current position
    - Otherwise, add step in opposite direction
- Is this planner complete?
  - Without obstacles?
  - With obstacles?



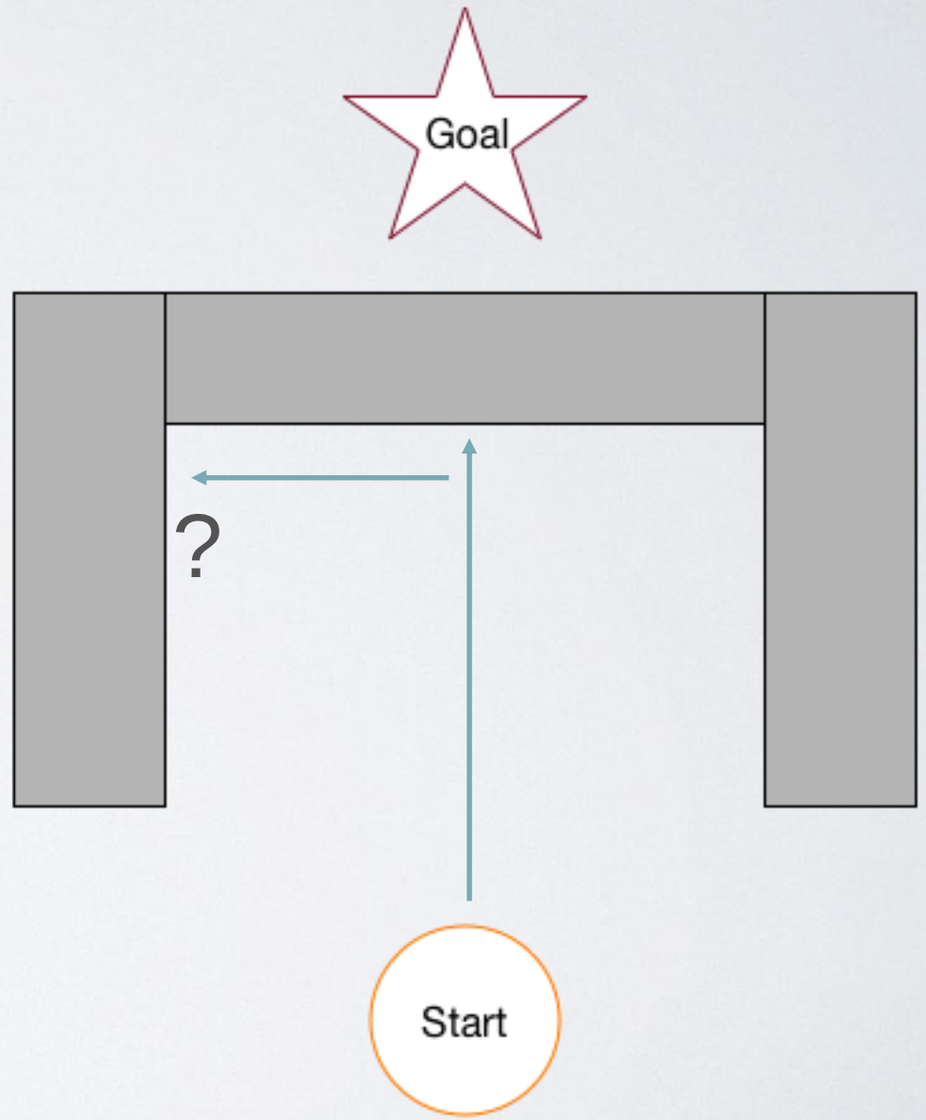
2D PATH PLANNING

- Can we use the algorithm in 2D?
- Introduces issue of ties
  - How can we resolve them?
- Known as gradient descent or potential field planning



POTENTIAL FIELDS

- Cul-de-sac still causes problems
- Always taking best immediate action
- Need to be less *greedy*
- Note: these algorithms assume ability to measure distance to goal

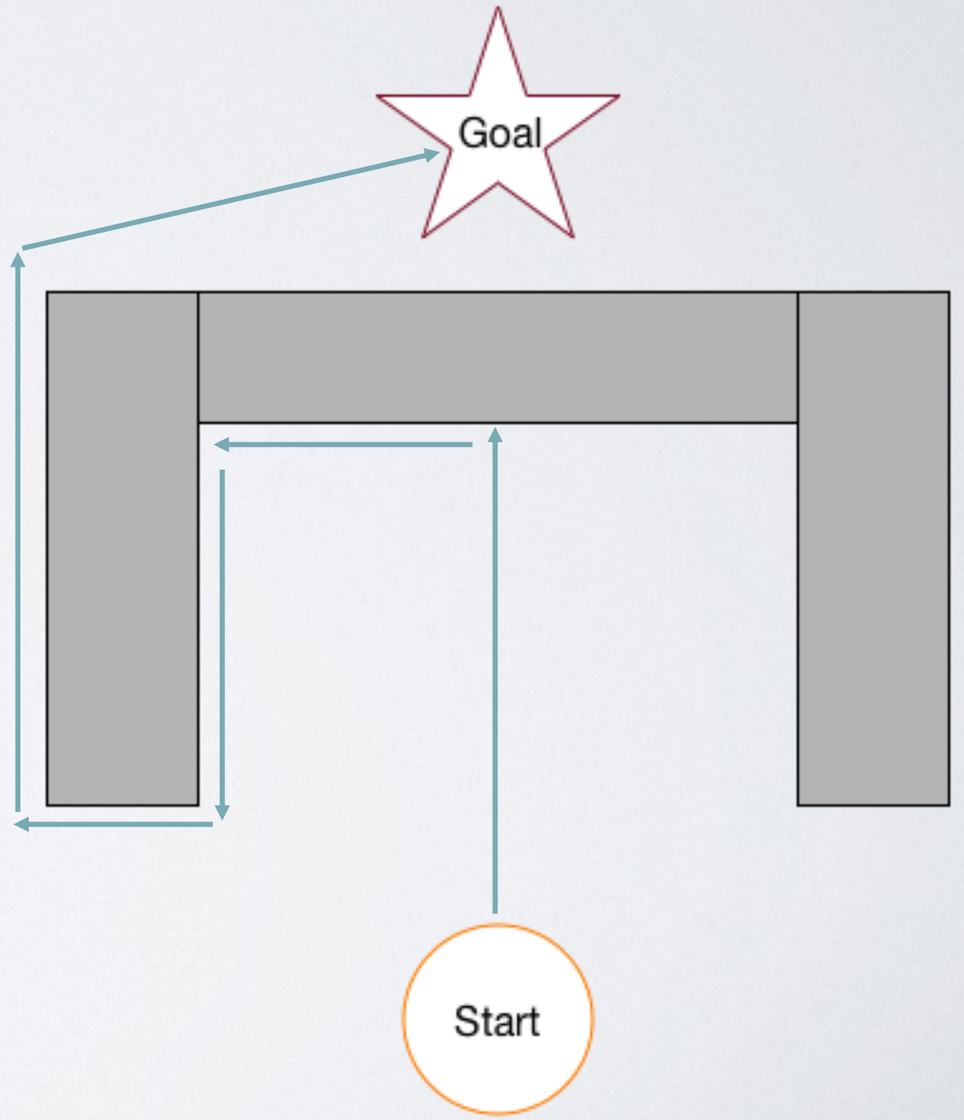


BUG 0

- Assume
  - Can always measure direction to goal
  - Can sense obstacle immediately adjacent to robot
  - Finite number of obstacles with finite size
- Can extend previous algorithm to handle obstacles

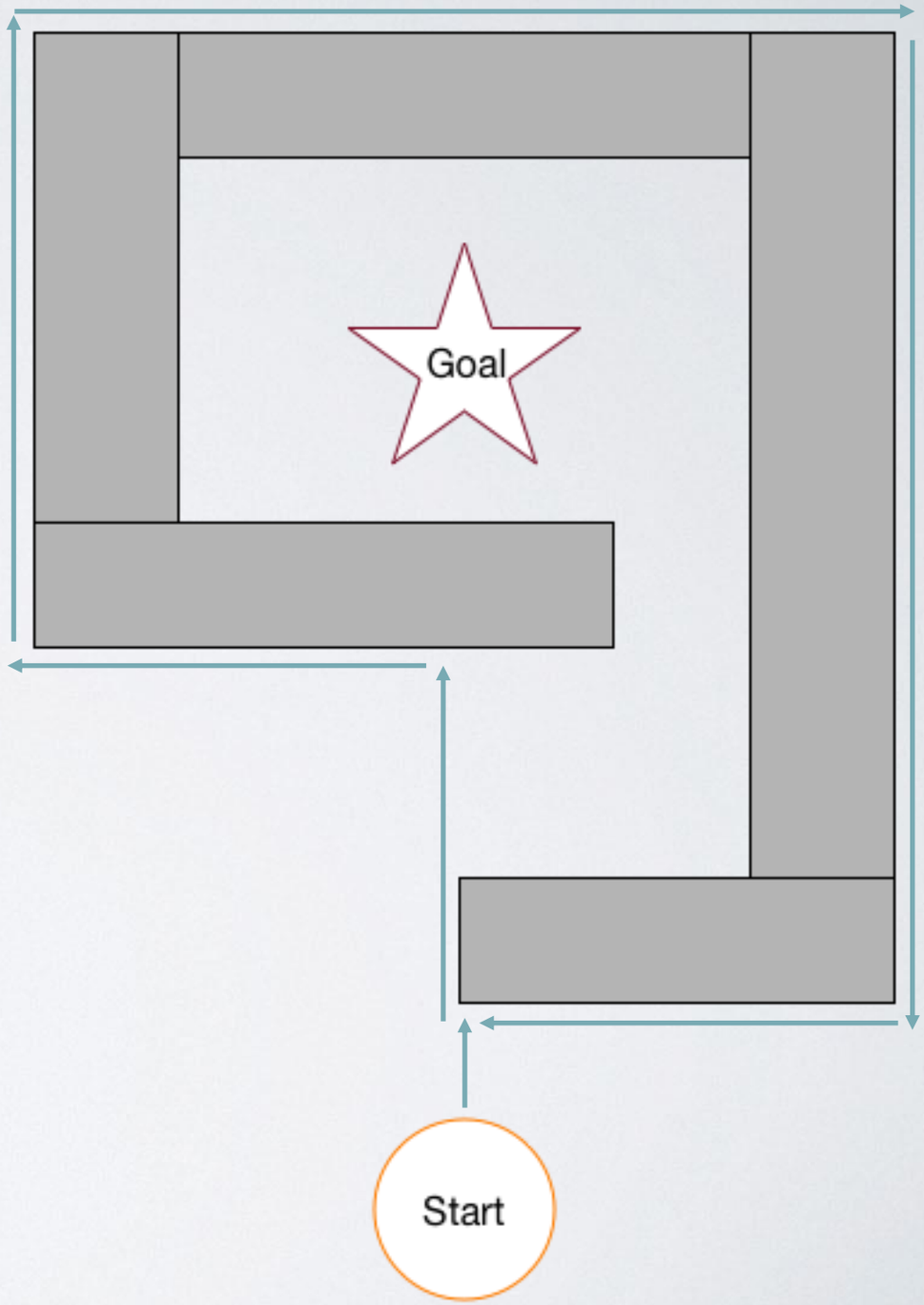
BUG 0

- Head towards goal
- When obstacles encountered, drive along boundary until path to goal is clear again
- Is this planner complete?



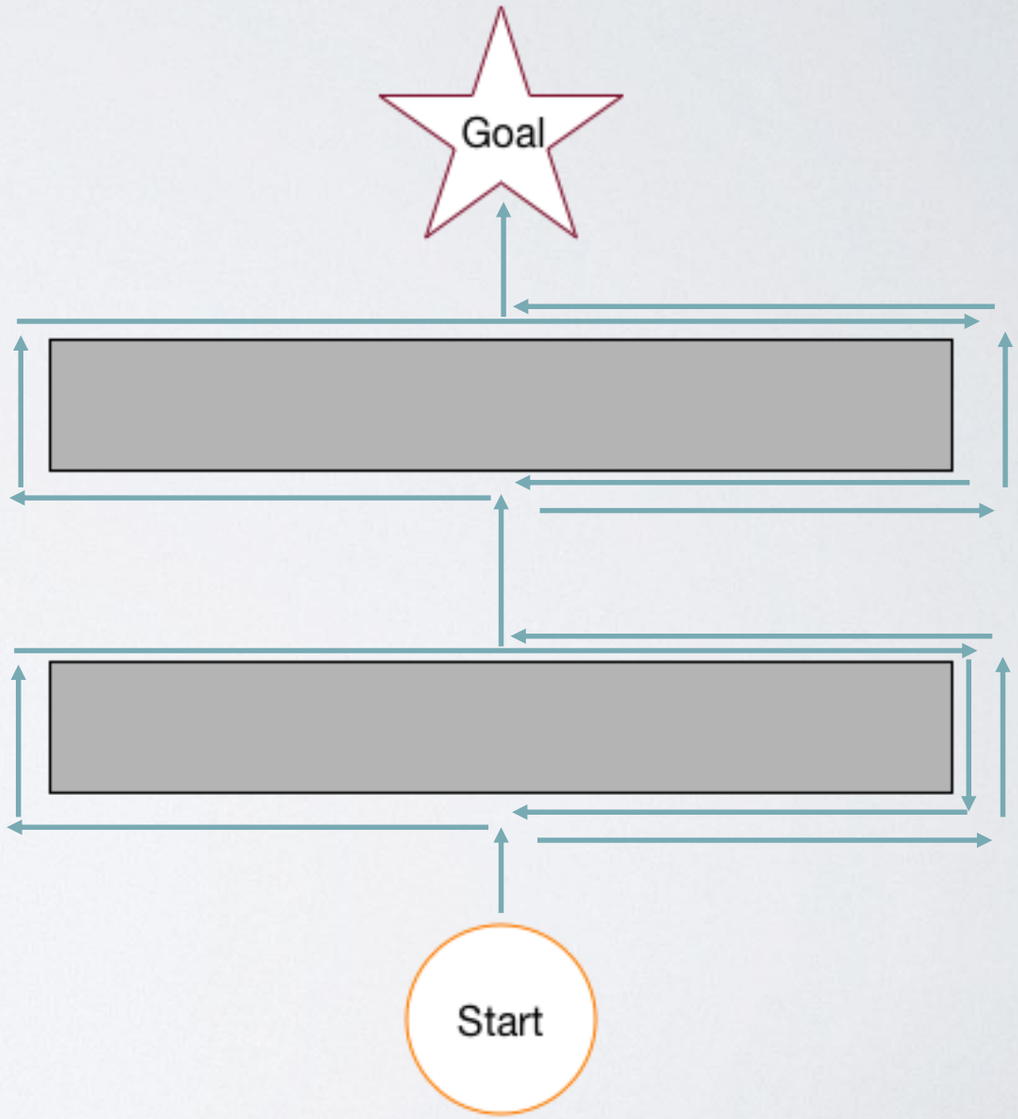
BUG 0 BUGS

- Algorithm can loop forever
- How to solve this?
  - Coin flipping can help
  - Memory solves it



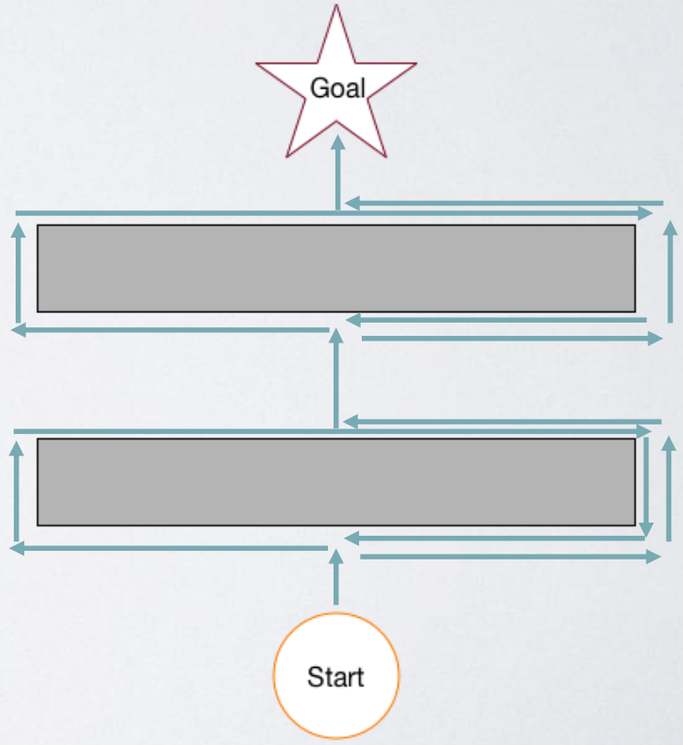
BUG 1

- Assume measuring distance to goal
- Head towards goal
- When obstacles encountered, circumnavigate obstacles
  - Record point on obstacle where goal was closest
  - Once circumnavigation complete, return to recorded point and start over
  - If obstacle is re-encountered, fail
- Is this complete?



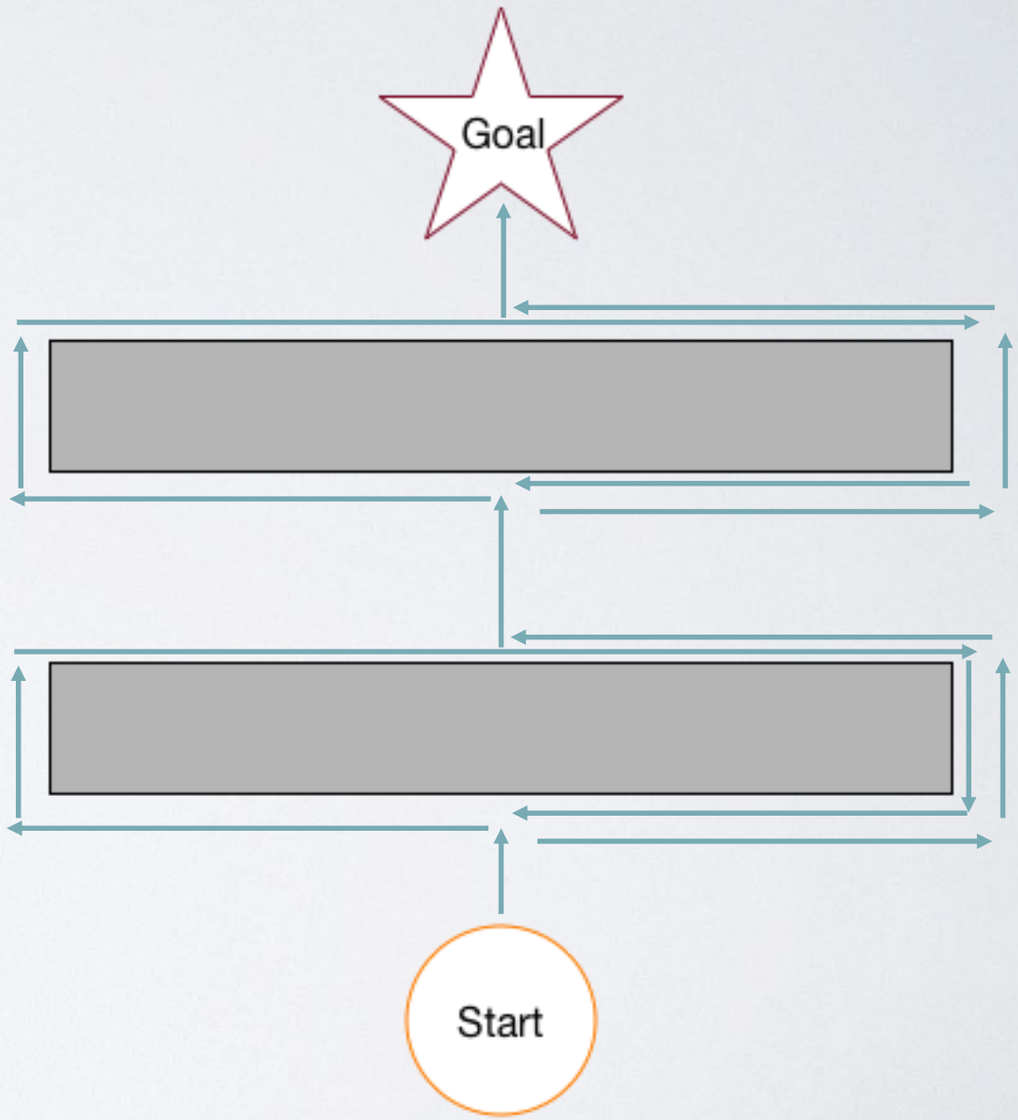
BUG 1

- Bug 1 is *provably* complete
- Proof by contradiction. Assume path exists, if:
  - Never terminates
    - Algorithm leaves every obstacle close than it arrives
    - Finite obstacles in world
  - Assume incorrect termination
    - Leaving an obstacle must cause hitting the same obstacle
    - Line from start to goal must enter an obstacle an even number of times
    - Every entrance to an obstacle has a corresponding point closer than entrance if a path exists



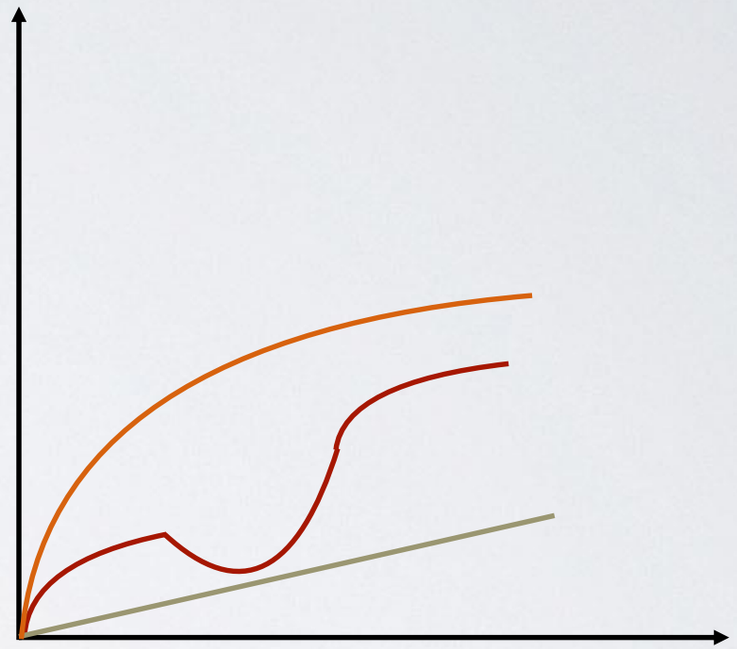
BUG 1 PERFORMANCE

- Bug 1 is *complete*
- Can also say something about the length of the solution found
- Concept of *bounds*



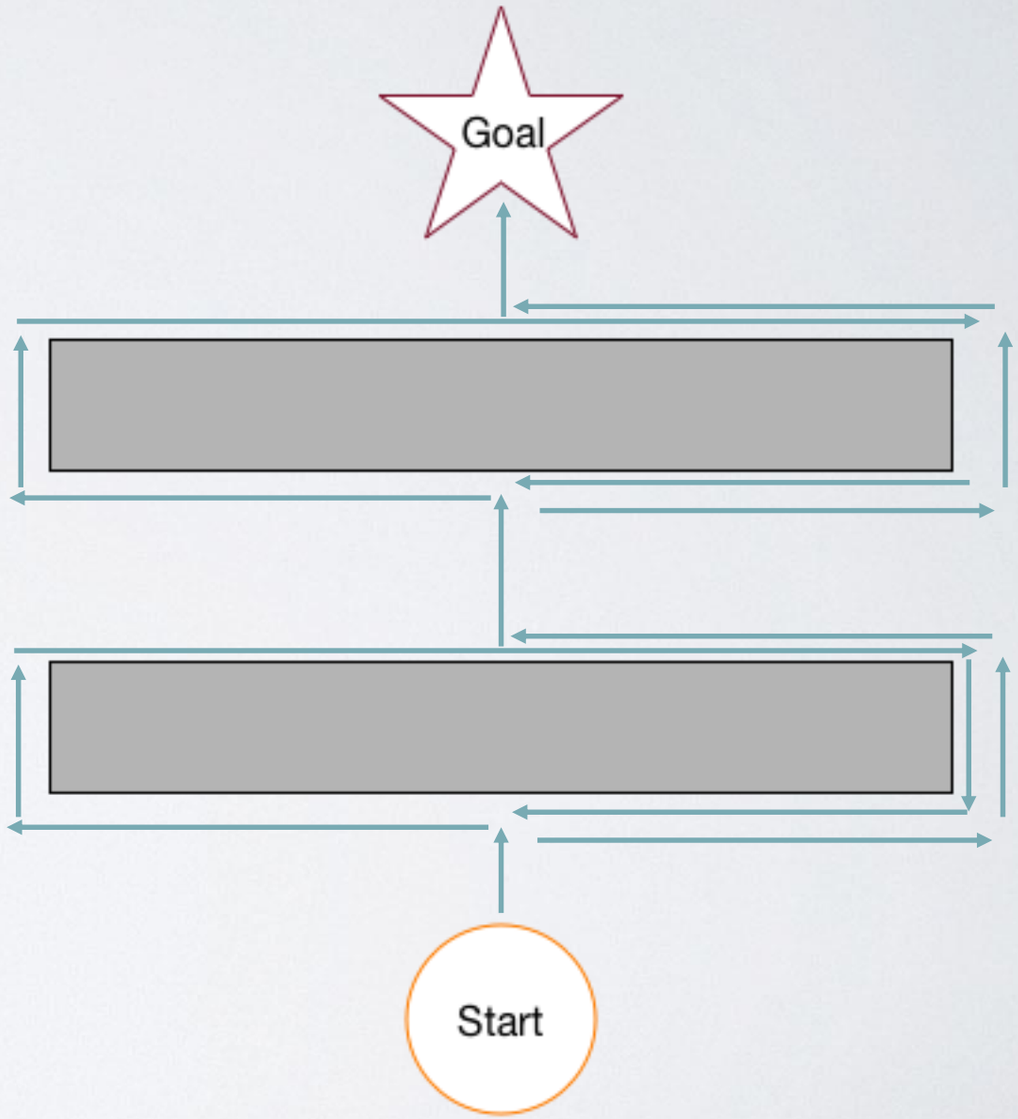
BOUNDS

- Estimate range of some function as inputs scale
  - Provable equations
- Provide worst and best case estimates



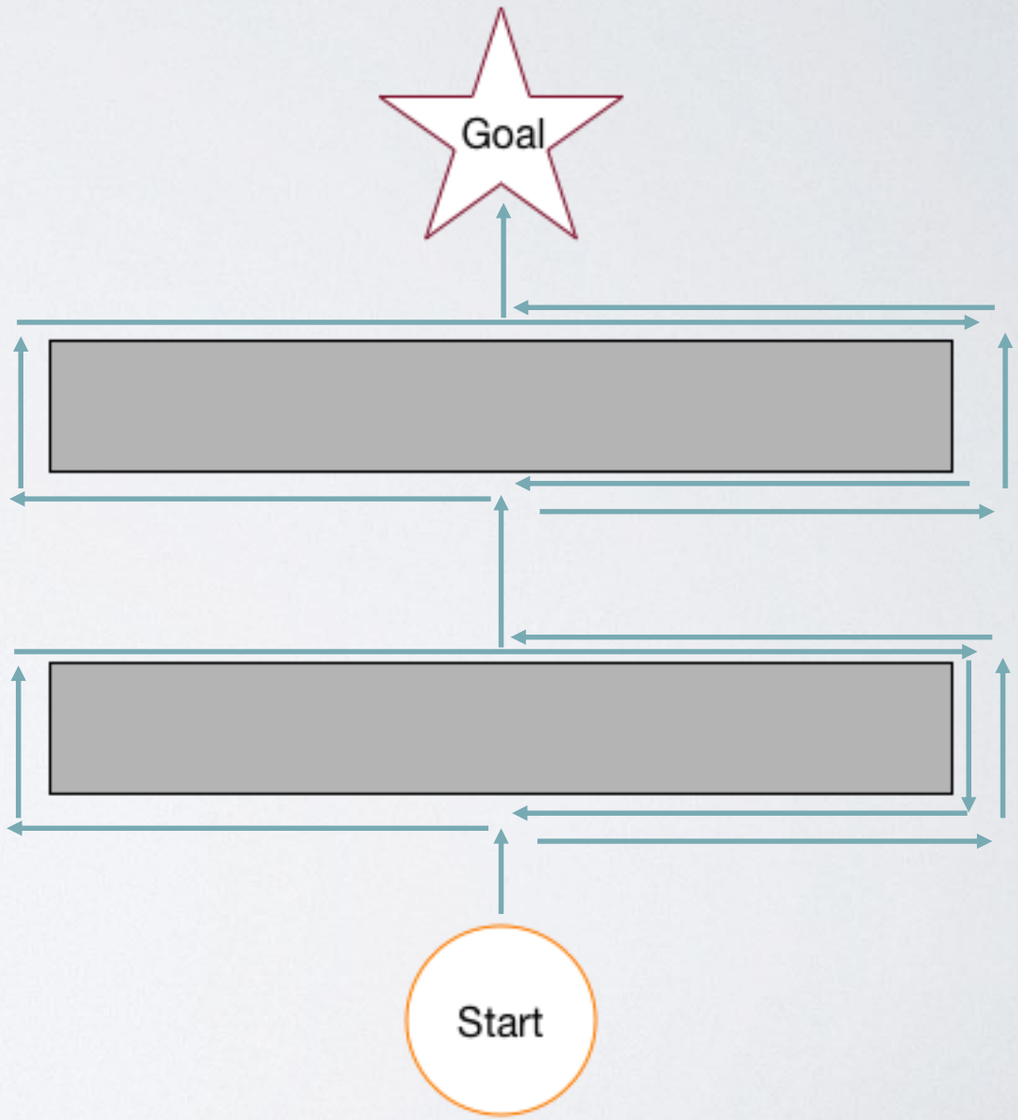
BUG 1 PERFORMANCE

- If straight line from start to goal is D
- Solution intersects N obstacles
- Lower bound?



BUG 1 PERFORMANCE

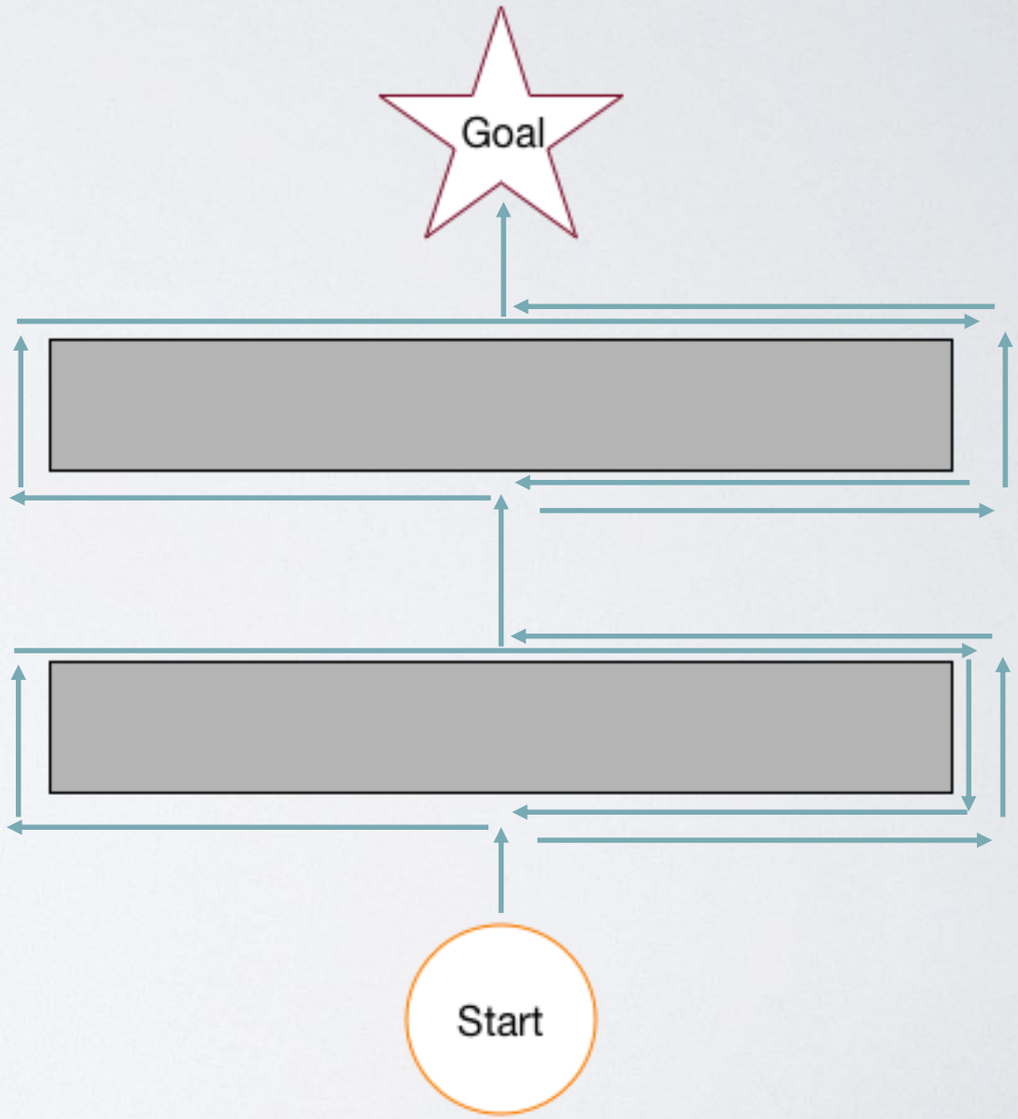
- If straight line from start to goal is D
- Solution intersects N obstacles
- Lower bound?
  - D
- Upper bound?



BUG 1 PERFORMANCE

- If straight line from start to goal is D
- Solution intersects N obstacles
- Lower bound?
  - D
- Upper bound?

$$D + \sum_{i=0}^{N} \frac{3}{2} P(O_i)$$



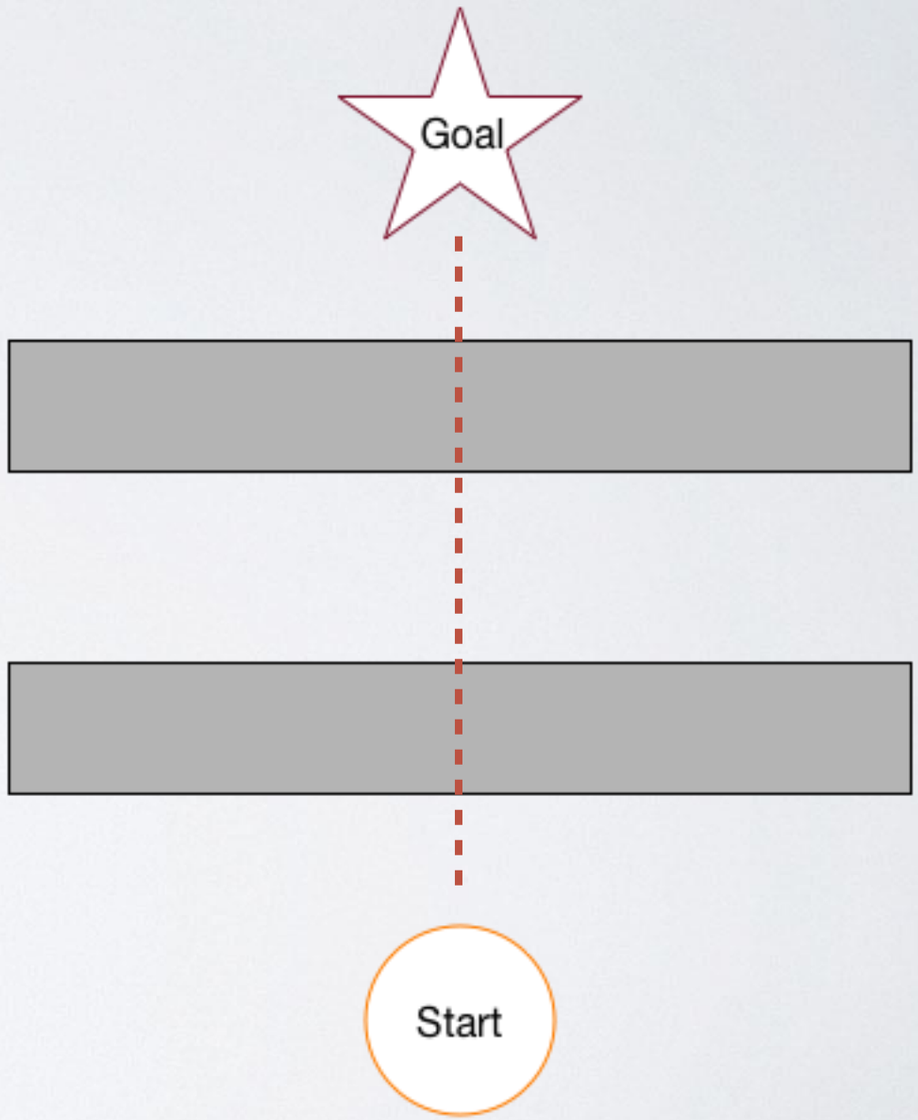
VirginiaTech
Invent the Future®

IS THERE BETTER?

- Bug 1 samples entire object perimeter
- Reasonable criteria for leaving obstacle early?
- Need some heuristic for leaving obstacle early

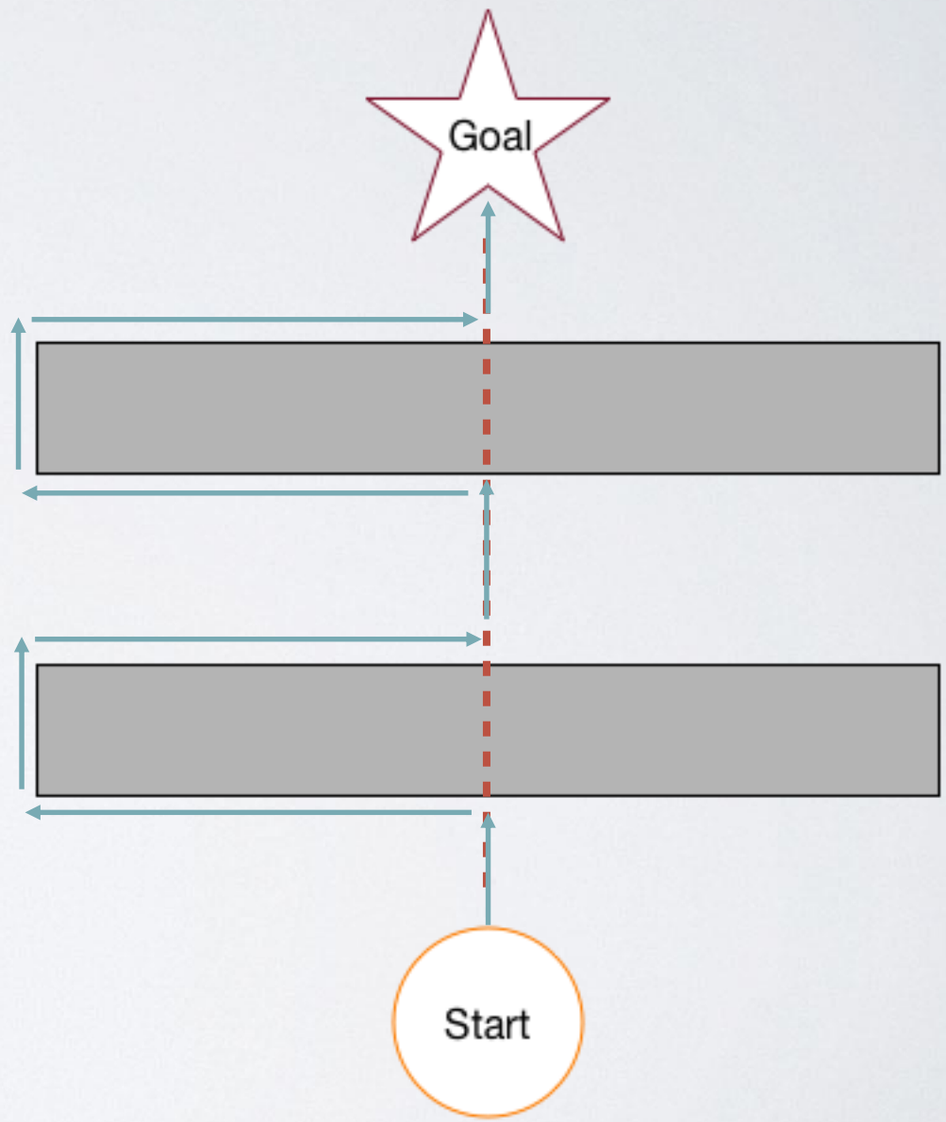
IS THERE BETTER?

- Bug 1 samples entire object perimeter
- Reasonable criteria for leaving obstacle early?
- Need some heuristic for leaving obstacle early
- Straight Line Path



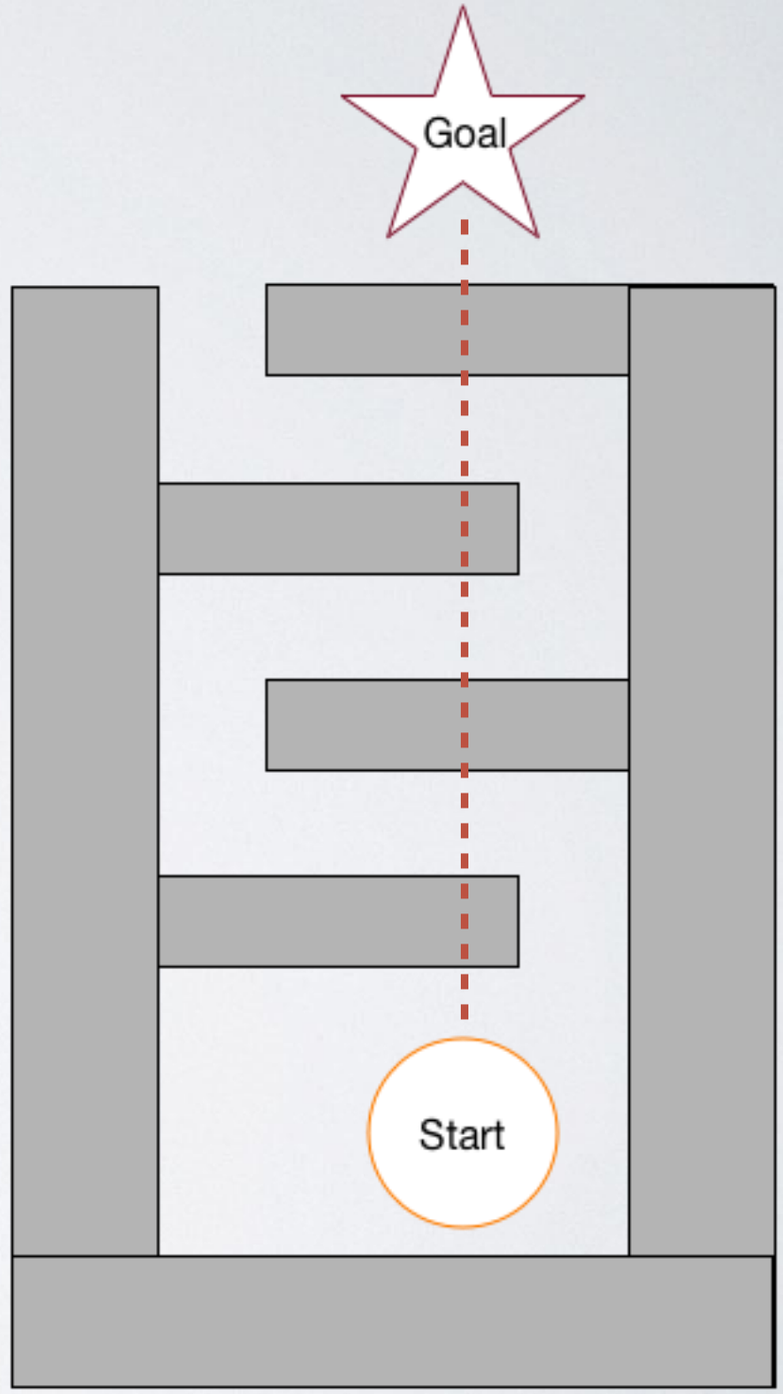
BUG 2

- Head towards goal
- Navigate around obstacles
  - If a point on the straight line to the goal is encountered closer than original point, start over
- Is this always a better algorithm?



BUG 2

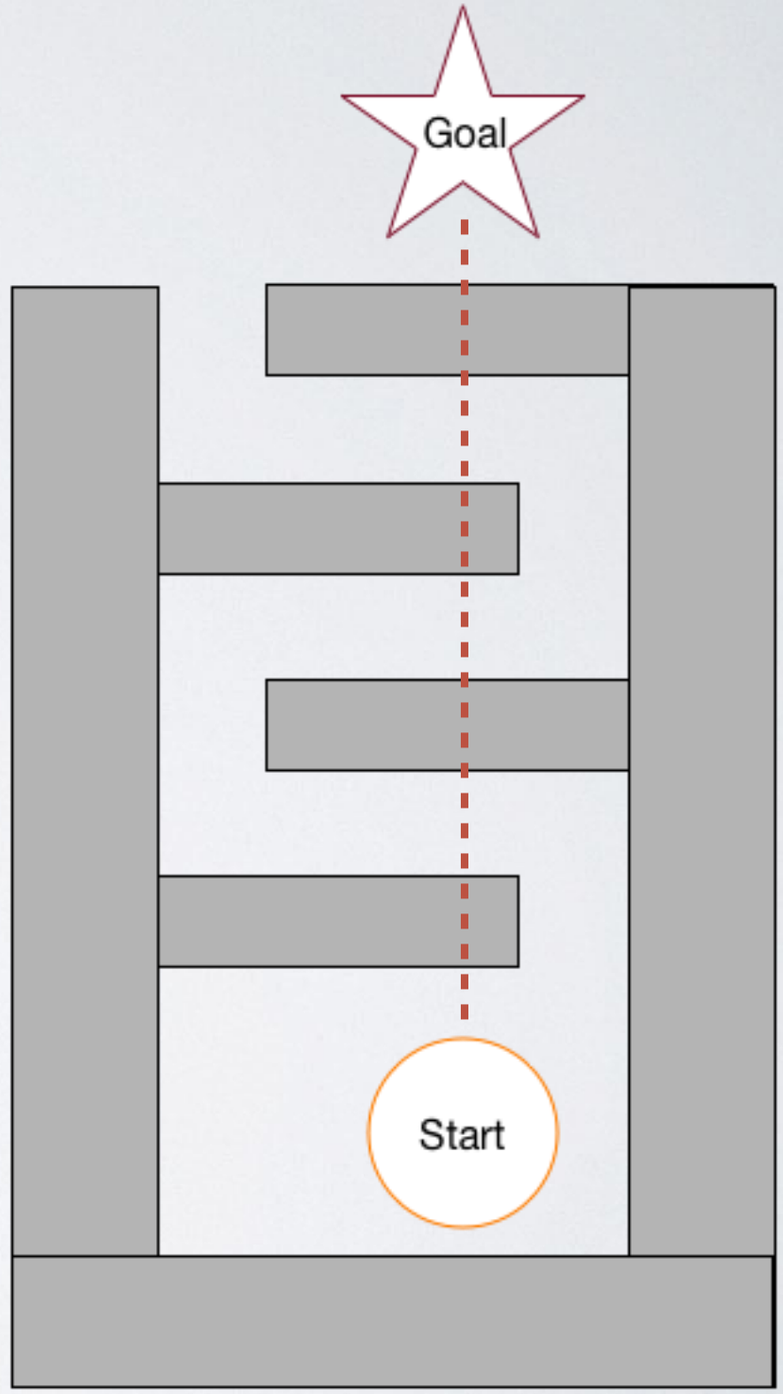
- Empirically, often performs better
- Depends on the environment
- What's the upper bound?



BUG 2

- Empirically, often performs better
- Depends on the environment
- What's the upper bound?

$$D + \sum_{i=1}^{N} \frac{X_i}{2} P(O_i)$$



VirginiaTech
Invent the Future®

LOCAL PLANNER REVIEWS

- Potential fields are fast
  - Can’t handle complex obstacles
- Bug 0 is more robust, but not complete
- Bug 1 is complete
  - Requires some memory
  - Tests full perimeter of every obstacle encountered
- Bug 2 can be better
  - Can suffer from pathological cases

LOCAL PLANNER PROBLEMS

- These algorithms rely on local sensing
- Integrate action with planning
- How to use ranged sensing?
- Deliberative planning?