Day 13

Fundamental Problems in Mobile Robotics

Fundamental Problems

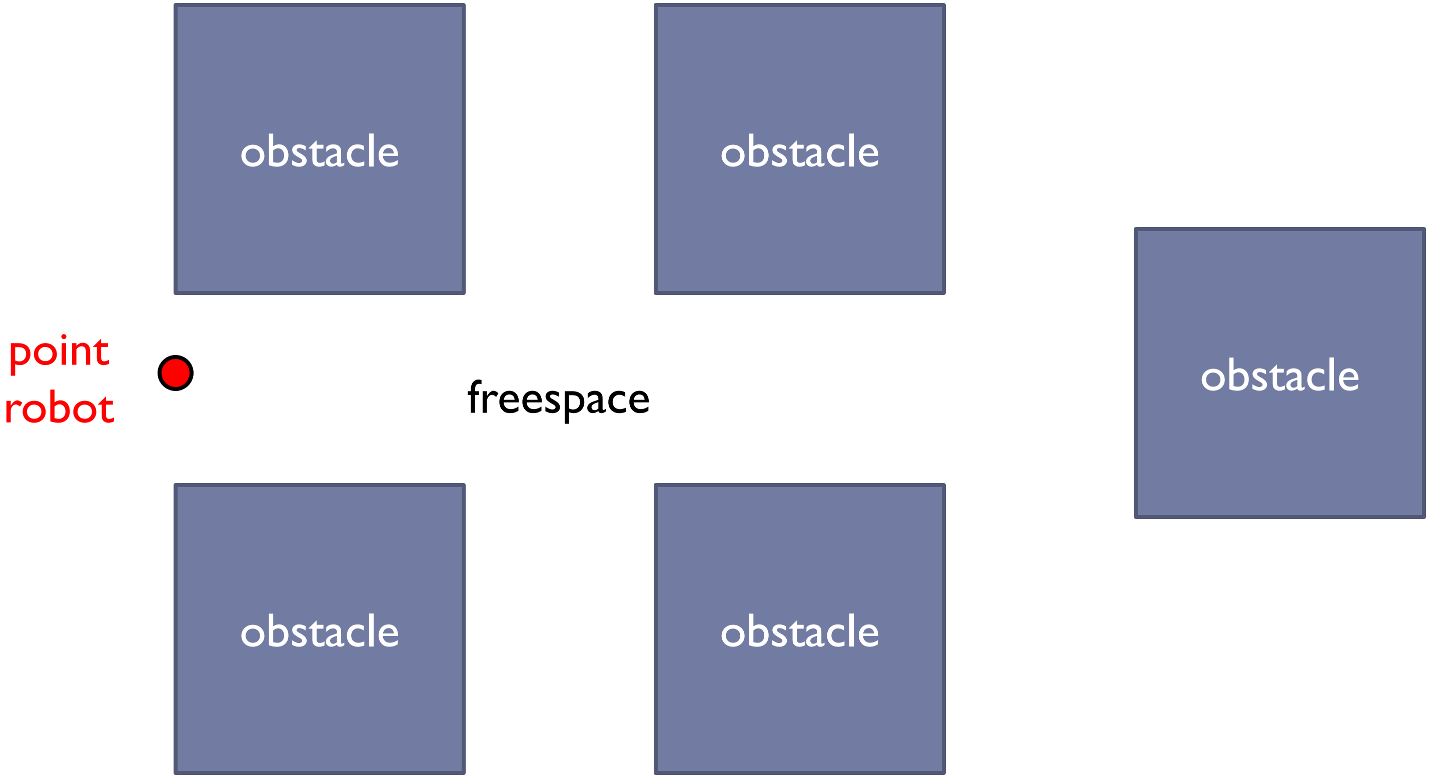
- Chapter 2 of Dudek and Jenkin begins:
  - "Before delving into the harsh realities of real robots..."

A Point Robot

- represents a mobile robot as a point in the plane*
- the point P fully describes the state of the robot
  - called pose or configuration
- robot motion causes the state to change
  - i.e., if the robot moves from P to Q then its state changes from

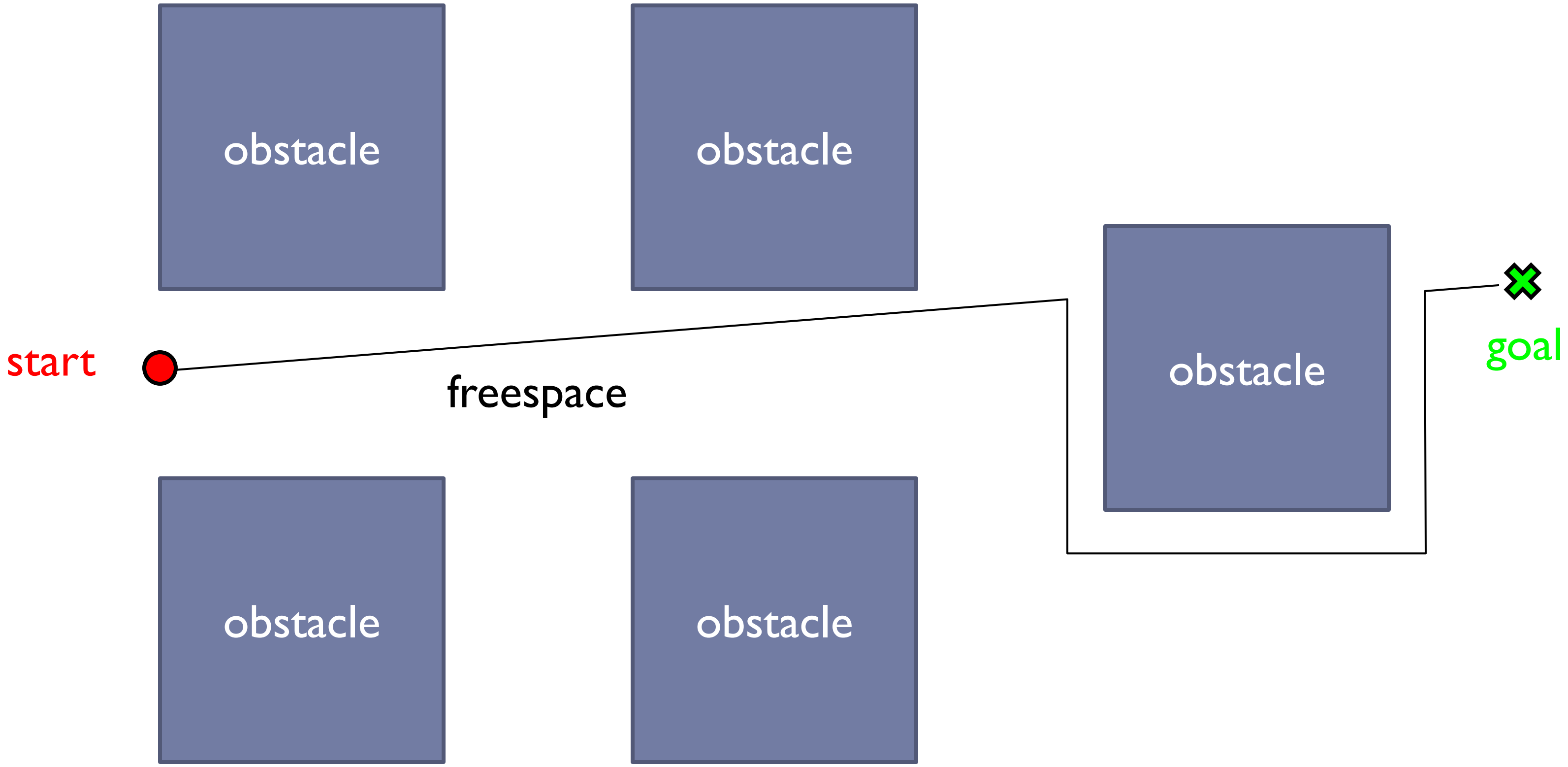
Free Space and Obstacles

- the set of valid poses is called the free space C_{free} of the robot
- the invalid poses are obstacles



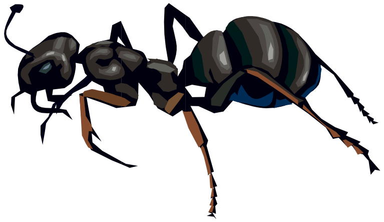
Path Planning

- is it possible for the robot to move to a goal configuration while remaining in C_{free} ?



Path Planning Using Bugs

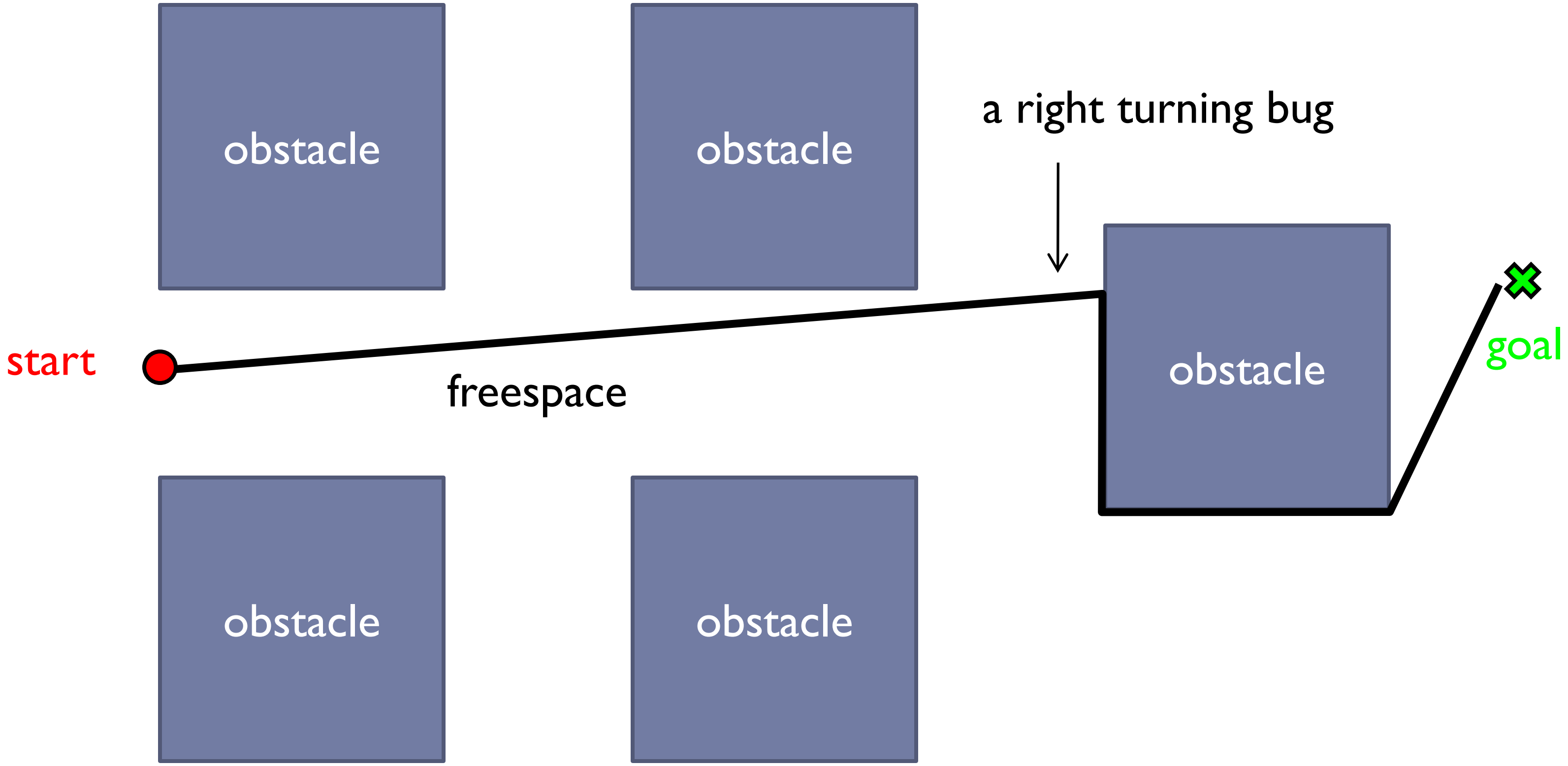
- bug algorithms assume:
  - point robot
  - known goal location
  - finite number of bounded obstacles
  - robot can perfectly sense its position at all times
  - robot can compute the distance between two points
  - robot can remember where it has been
  - robot can perfectly sense its local environment
  - robot can instantaneously change direction



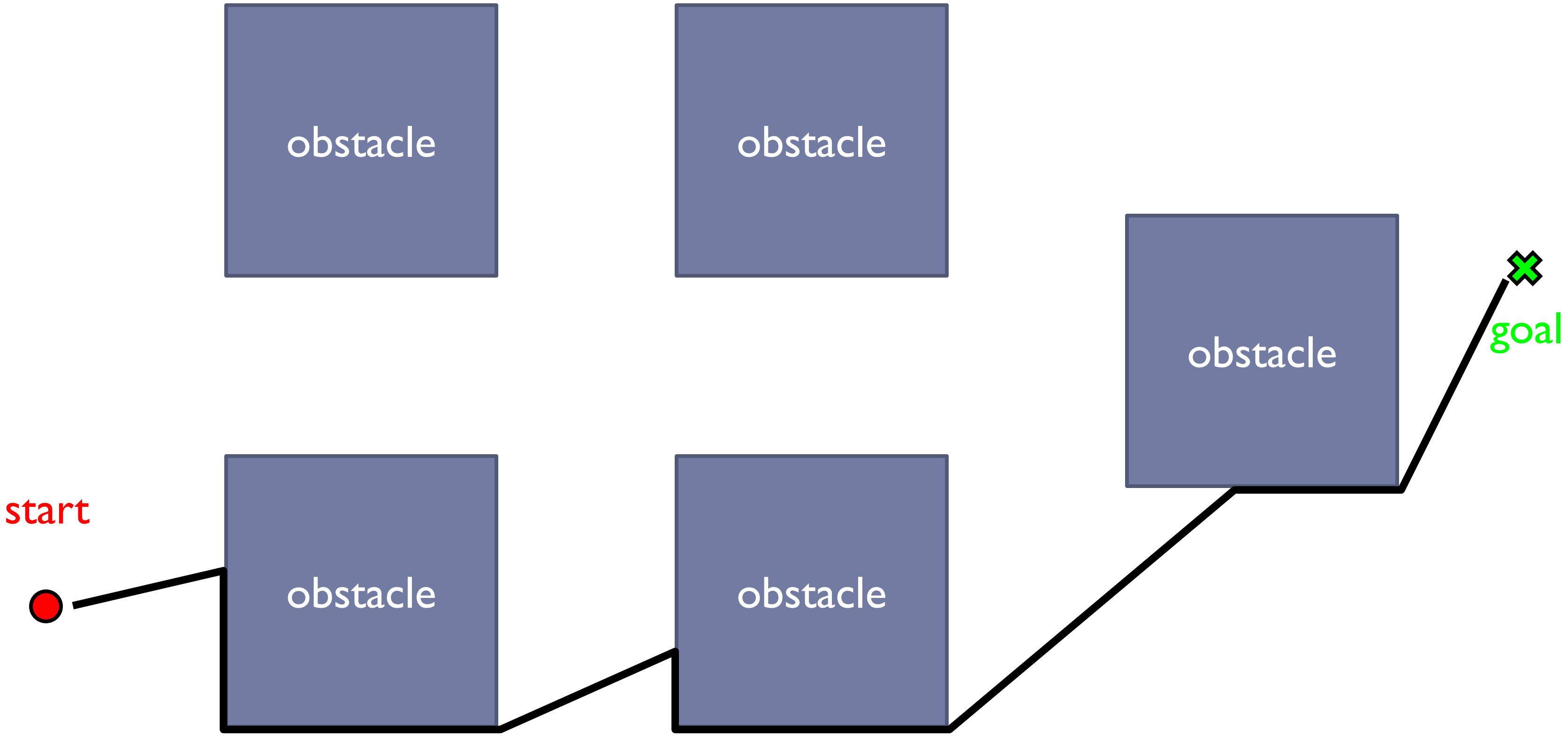
Bug Zero

- assumes a perfect contact sensor
- repeat
  - head towards goal
  - if goal is reached then stop
  - if an obstacle is reached then follow the boundary until heading towards the goal is again possible

Bug Zero

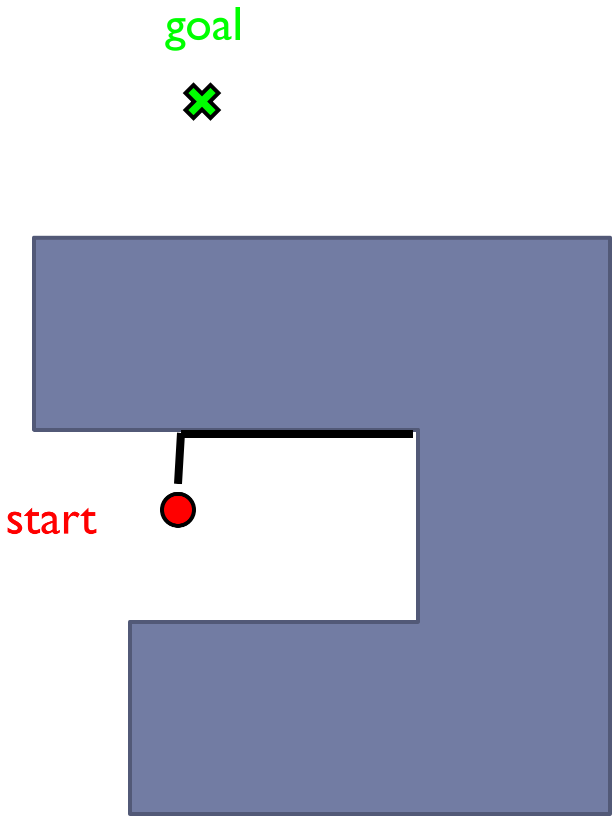

obstacle
obstacle
a right turning bug
start
obstacle
goal
freespace
obstacle
obstacle

Bug Zero



Bug Zero

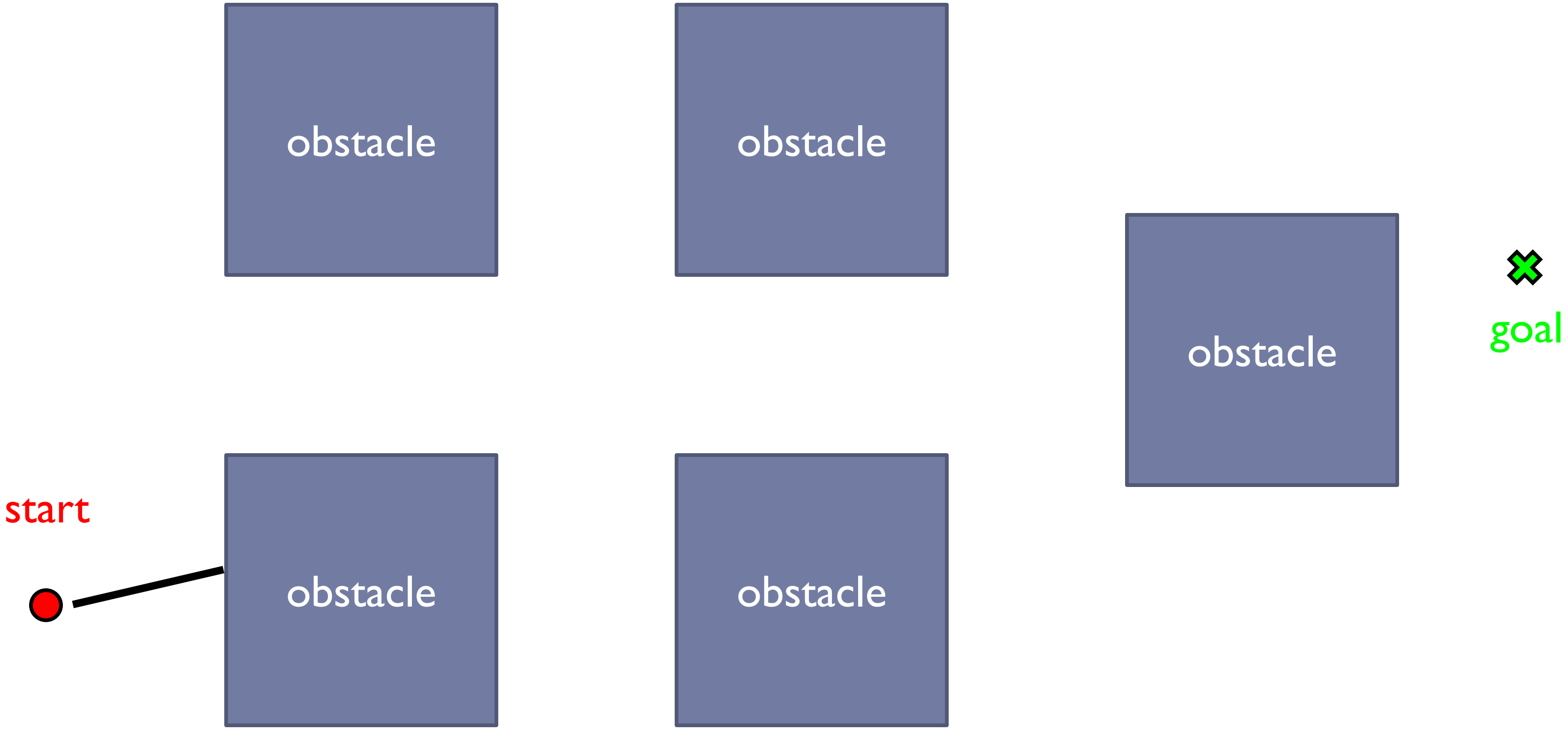
- not guaranteed to reach the goal



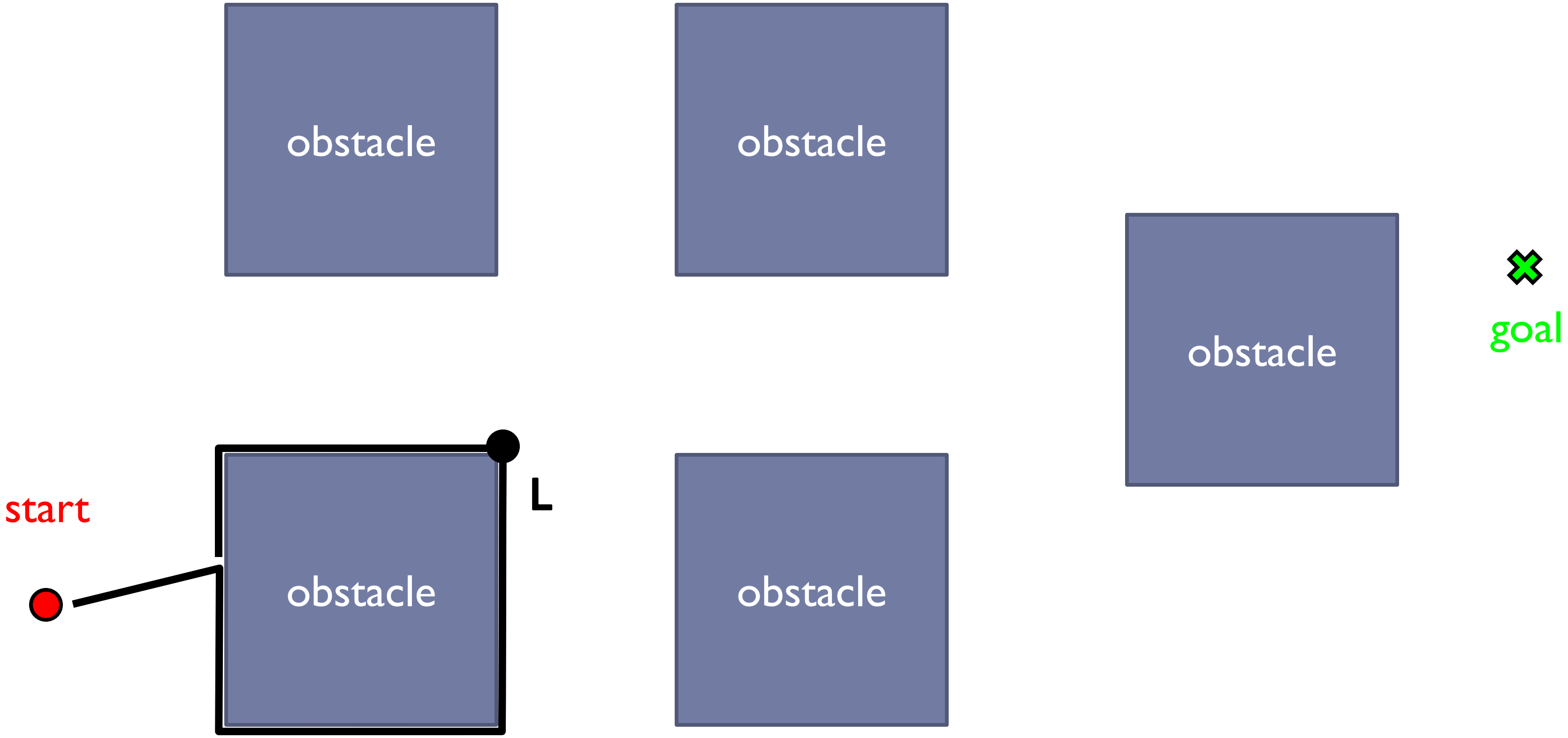
Bug One

- assumes a perfect contact sensor
- repeat:
  - head toward goal T
  - if goal is reached then stop
  - if an obstacle is reached then
    - remember the point of first contact H (the hit point)
    - follow the boundary of the obstacle until returning to H and remember the point L (the leave point) closest to T from which the robot can depart directly towards T
      - □ if no such point L exists then the goal is unreachable; stop
    - move to L using the shortest boundary following path

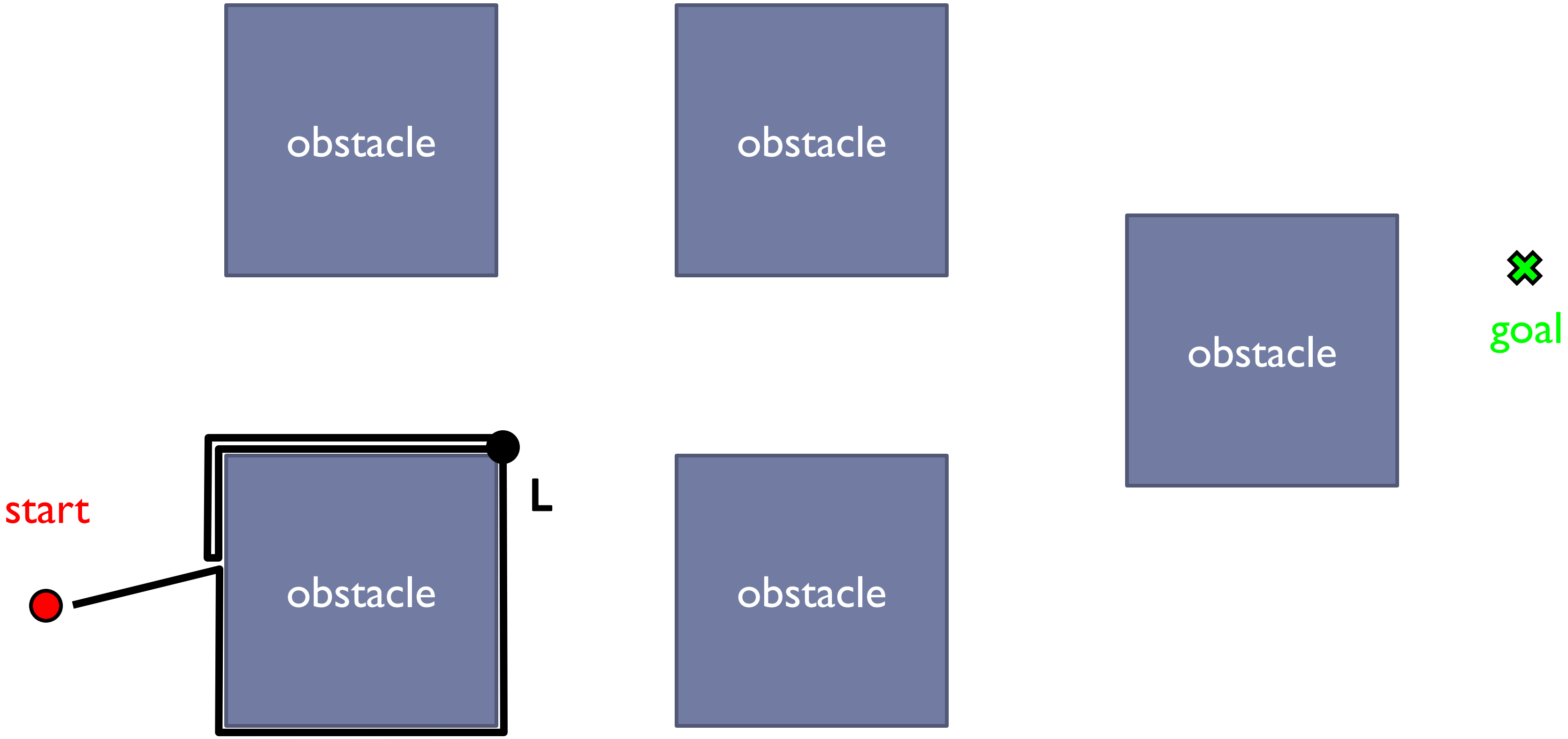
Bug One



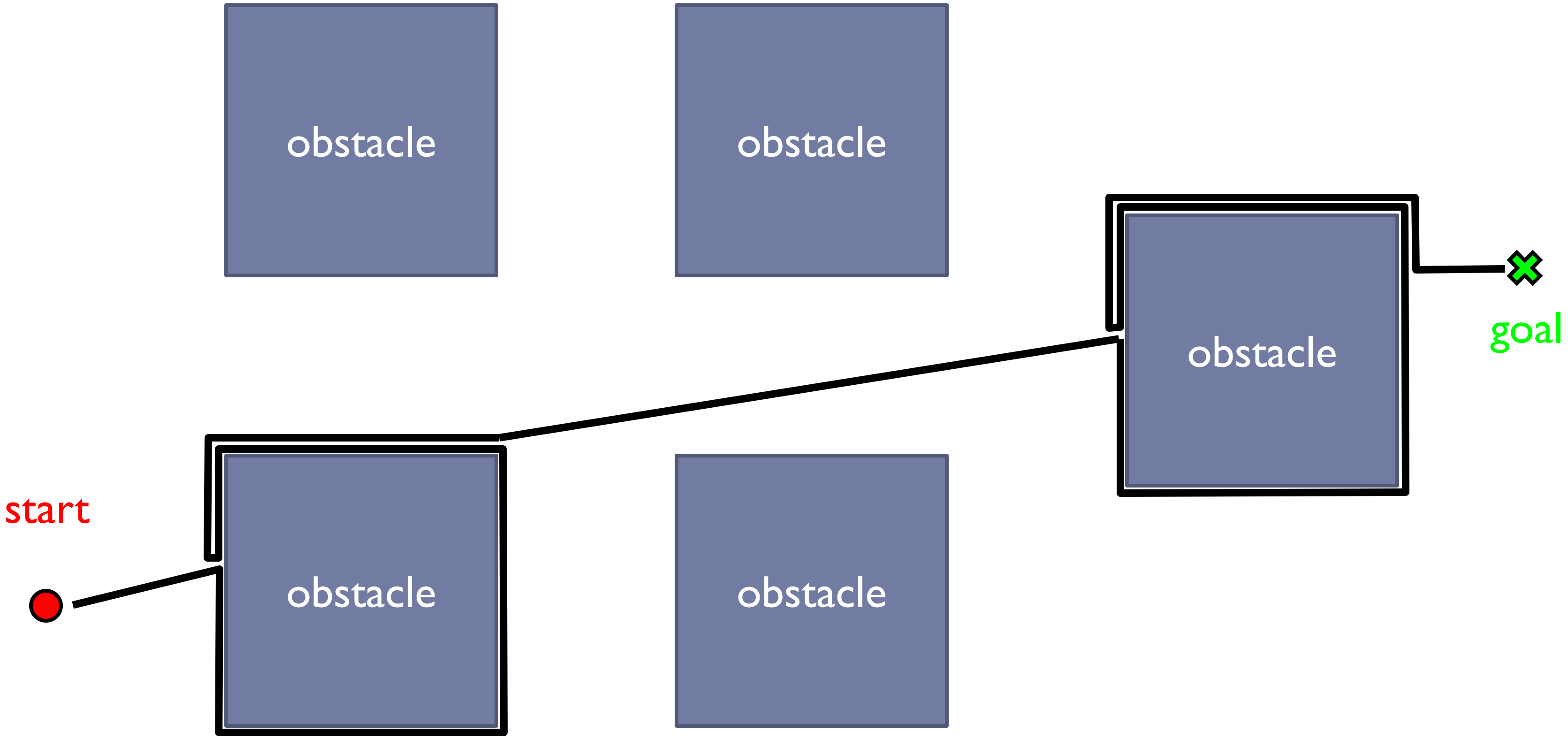
Bug One


obstacle
obstacle
obstacle
goal
start
L
obstacle
obstacle

Bug One

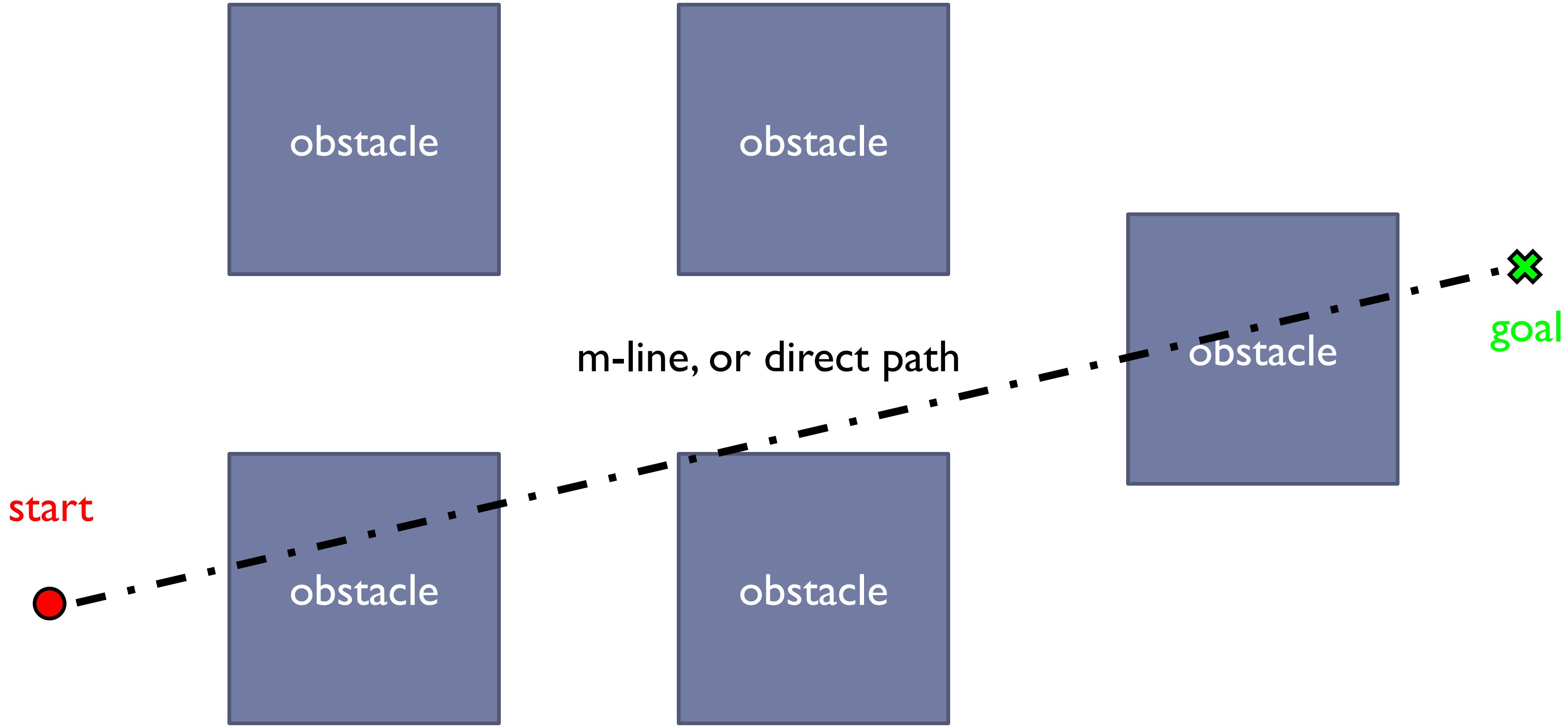

obstacle
obstacle
obstacle
goal
L
start
obstacle
obstacle

Bug One



Bug Two

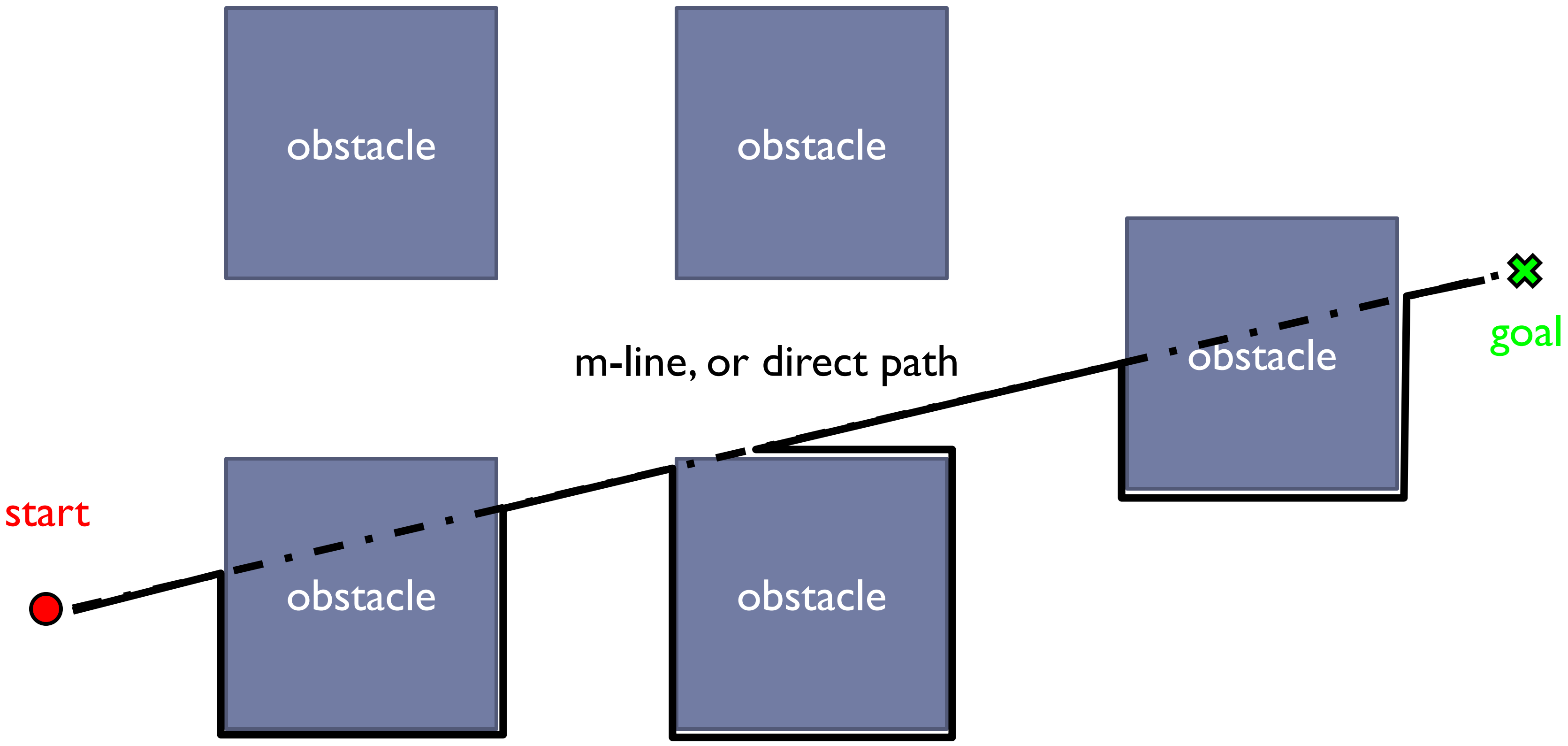
- Bug Two uses a line, called the *m-line*, from the start point to the goal
  - textbook calls the m-line the *direct path*



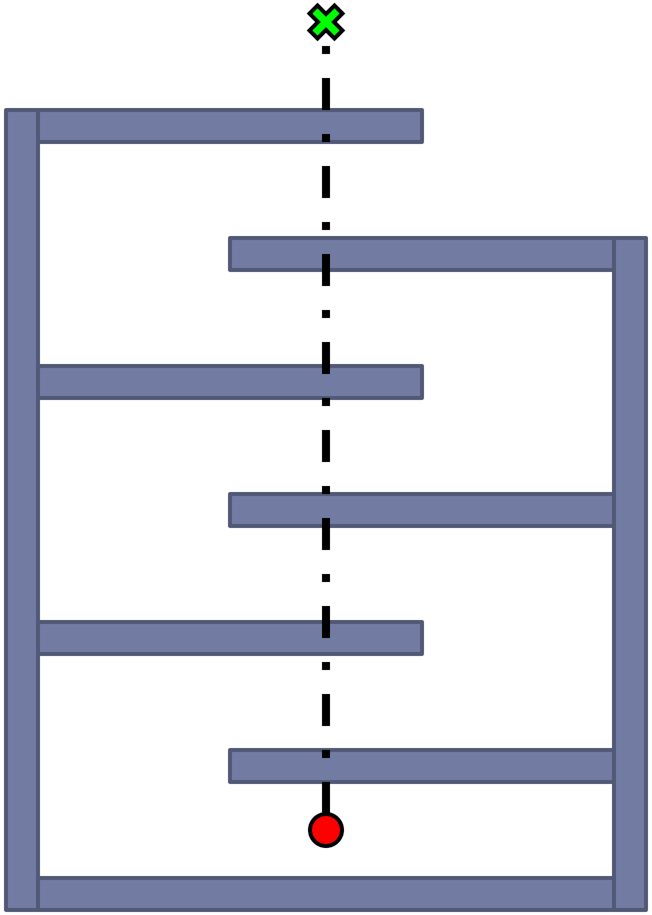
Bug Two

- assumes a perfect contact sensor
- repeat:
  - head toward goal T along the m-line
  - if goal is reached then stop
  - if an obstacle is reached then
    - remember the point of first contact H (the hit point)
    - follow the boundary of the obstacle until the m-line is crossed at a leave point closer to the goal than H
      - □ if no such point L exists then the goal is unreachable; stop
    - leave the obstacle and head toward T

Bug Two


obstacle
obstacle
obstacle
m-line, or direct path
goal
start
obstacle
obstacle

Bug Two



Bug One versus Bug Two

- Bug One uses exhaustive search
  - it considers all leave points before leaving the obstacle
- Bug Two uses greedy search
  - it takes the first leave point that is closer to the goal