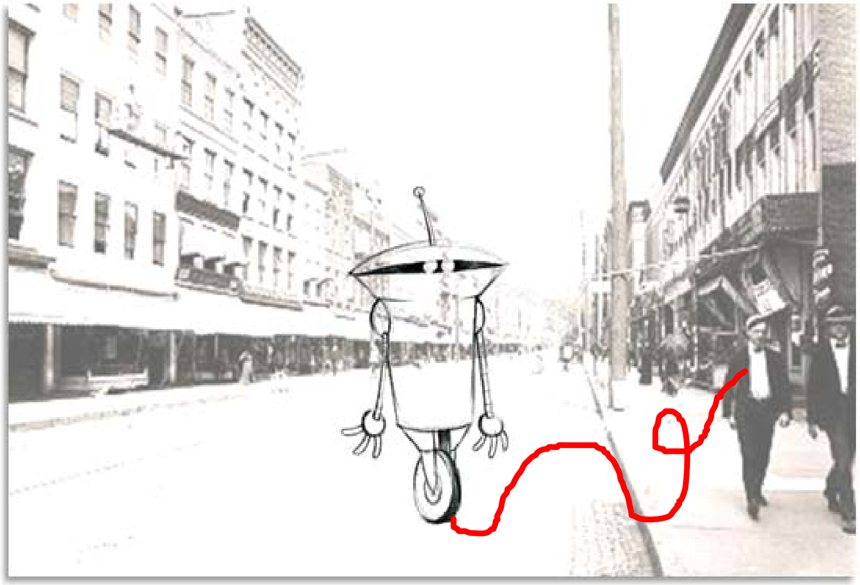
# Lecture 20: Motion planning (7)
# PRMs and RRTs



Topics:

- PRMs
- Challenging environments
- RRTs

Reading:

- Choset: 7
- LaValle: 5

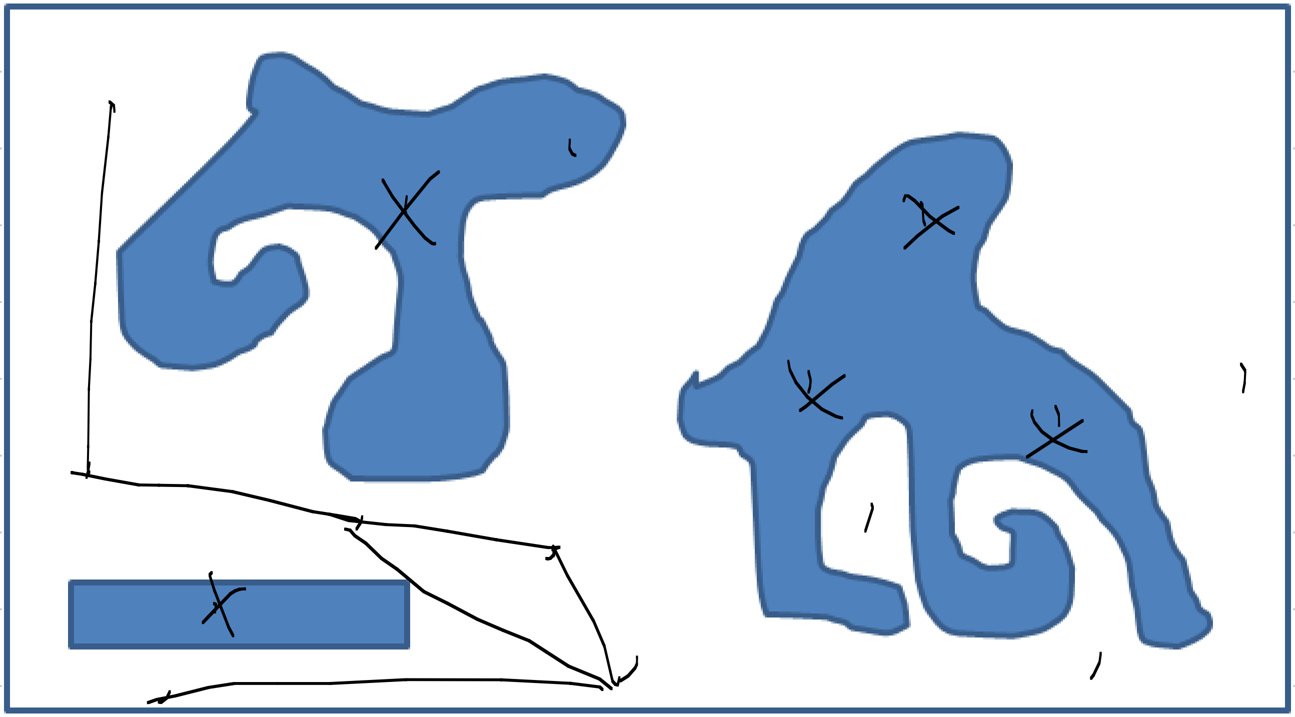
# Motion planning

Given: $q_{1:t}$, $q_{goal}$

sometimes: map, $q_0$

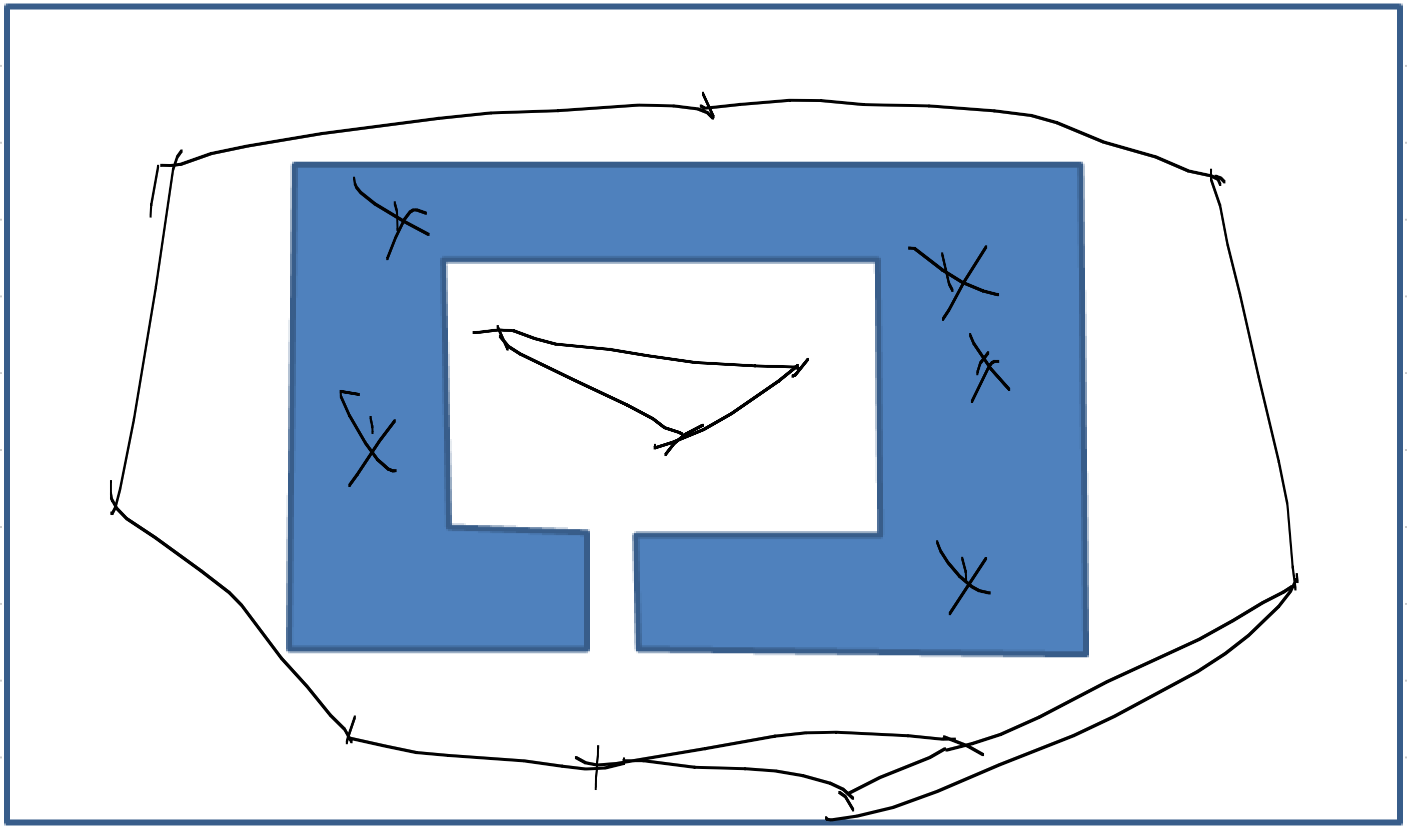
Find: $u_{1:t}$ s.t. $q_t = q_{goal}$

PRM :



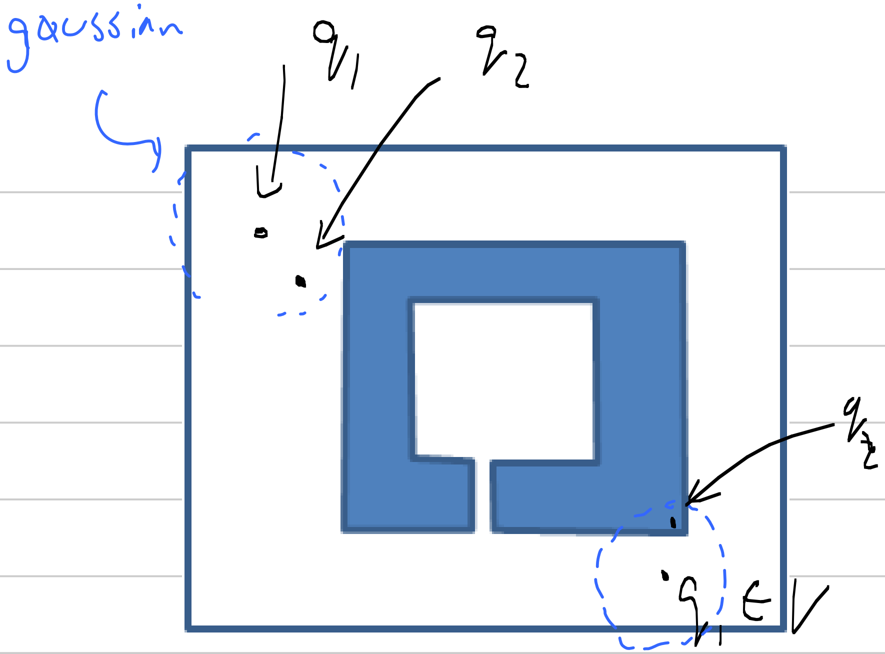
Sample → check → connect

problem: narrow passages → disconnected graph



Solution: Bias sampling

# Examples:



## 1) Gaussian sampler

Sample $q_1$

Sample $q_2 \sim N(q_1, \Sigma)$

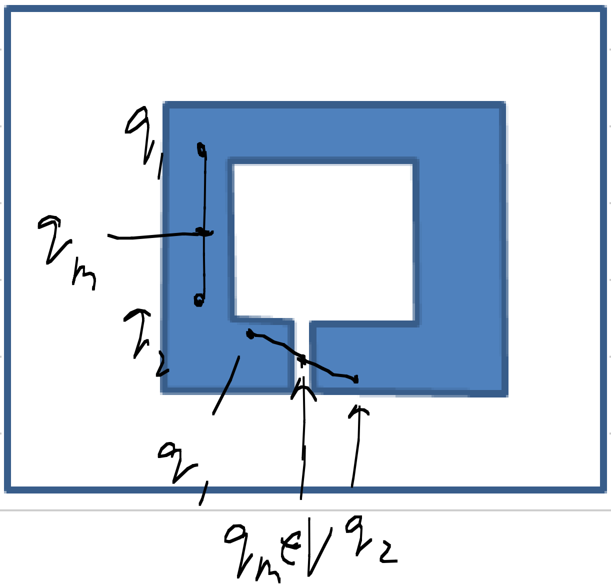
if $q_i \in Q_{free}$ and $q_j \in Q_{obs}$ $i, j \in \{1,2\}$, $i \neq j$

then $V = V \cup \{q_i\}$

## 2) Bridge planner

if $q_1, q_2 \in Q_{obs}$

if $q_m = \frac{q_1 + q_2}{2} \in Q_{free}$ then $V = V \cup \{q_m\}$



# Visibility-Based PRM

Ideally we want a sparse graph with good coverage

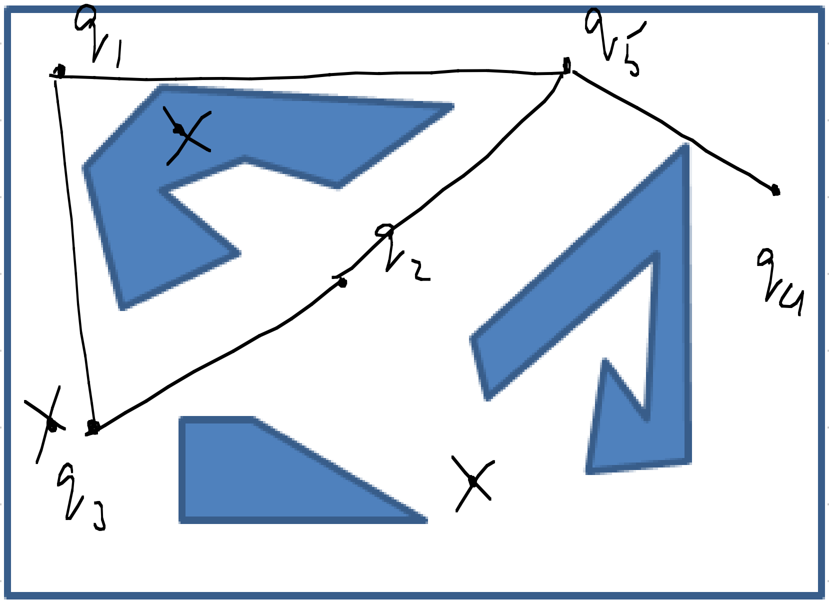
Visibility PRM:

PRM, except $q \in Q_{free}$ is added to V iff

$q$ cannot connect to any $q' \in V$

OR

$q$ can connect at least 2 unconnected components



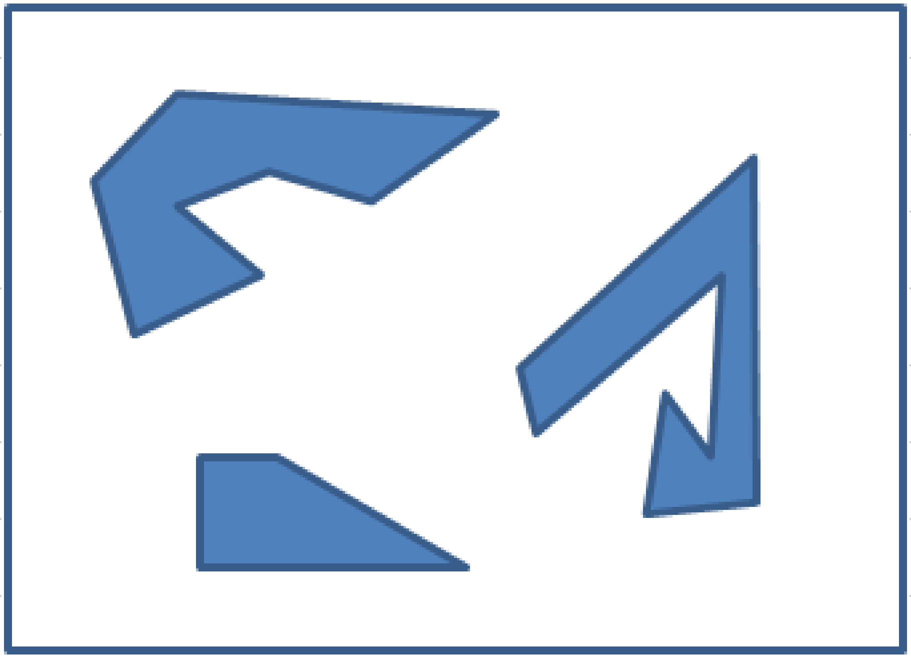
# Single queries

Given: $q_{start}$ - $q_{goal}$

ability to check $q \in Q_{free}$

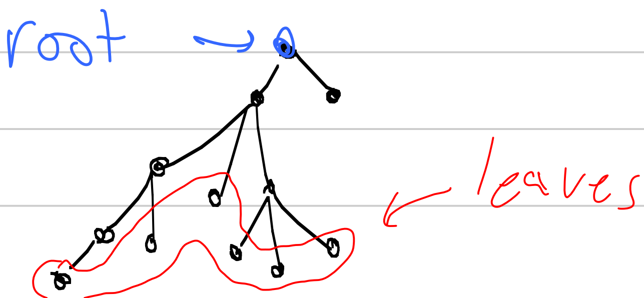
Find: path from $q_{start}$ to $q_{goal}$

$(q_{start}, q_{goal}) \in Q_{free}$

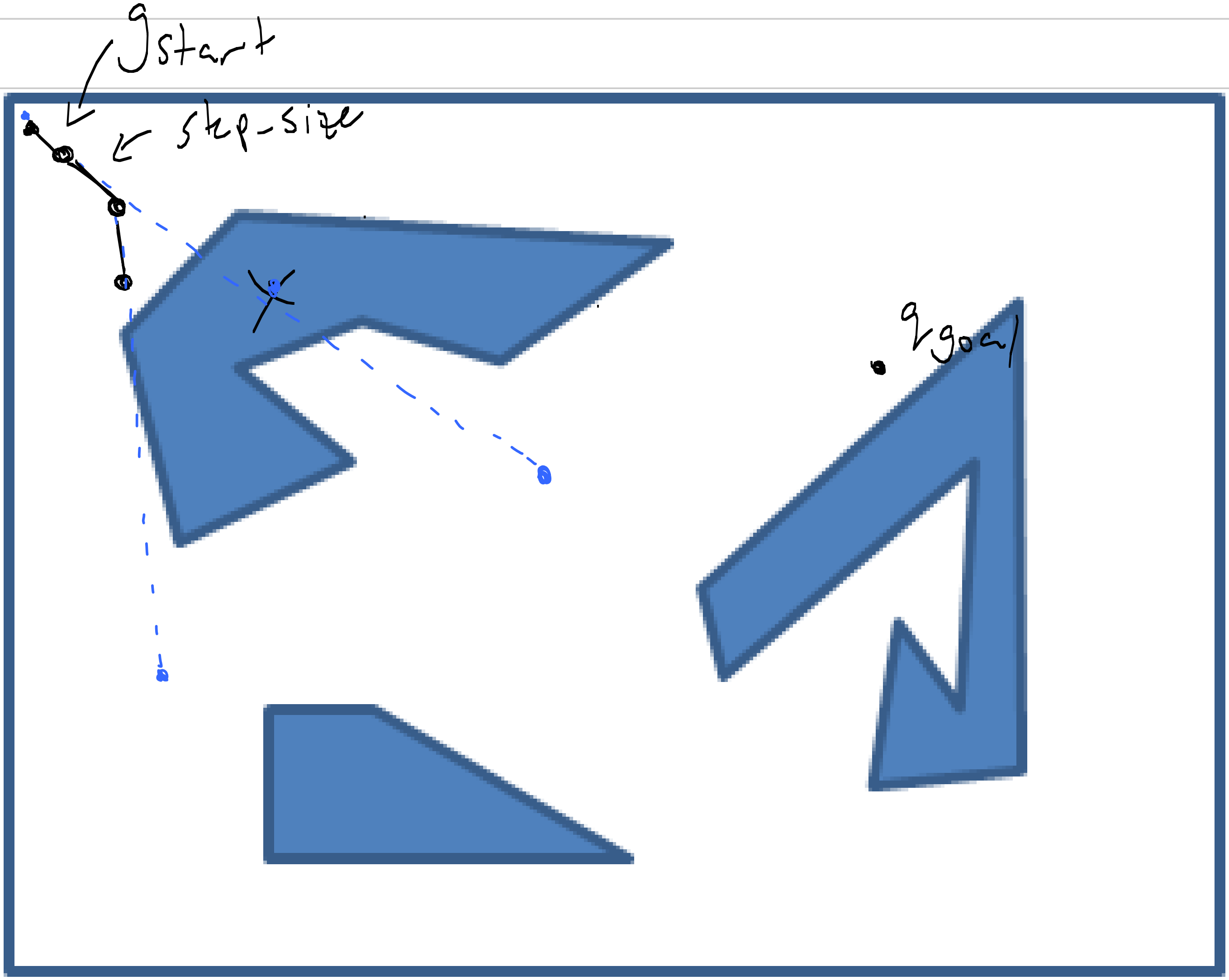


Idea: grow search tree(s)

Tree: a connected, a-cyclic graph (each node has one parent)



# Rapidly-Exploring Random Trees (RRT)



# RRT - algorithm

given: $q_{start}$, $q_{goal}$, check if $q \in Q_{free}$, step_size, $n$

find: $G=(V,E)$ a tree connecting $q_{start}$ and $q_{goal}$

Init: $V=\{q_{start}\}$ $E=\emptyset$

```
for $i = 1, \dots, n$
    - Sample $q_{rand} \in Q_{free}$
    - find $q_{near}$ : closest $q \in V$ to $q_{rand}$
    - $q_{new}$ = point on line $(q_{near}, q_{rand})$ that is
          step_size away from $q_{near}$
        if $q_{new} \in Q_{free}$ and $(q_{near}, q_{new}) \in Q_{free}$
```

then $V = V \cup \{q_{new}\}$, $E = E \cup \{(q_{near}, q_{new})\}$

- try to connect $q_{new}$ to $q_{goal}$ (other tree)

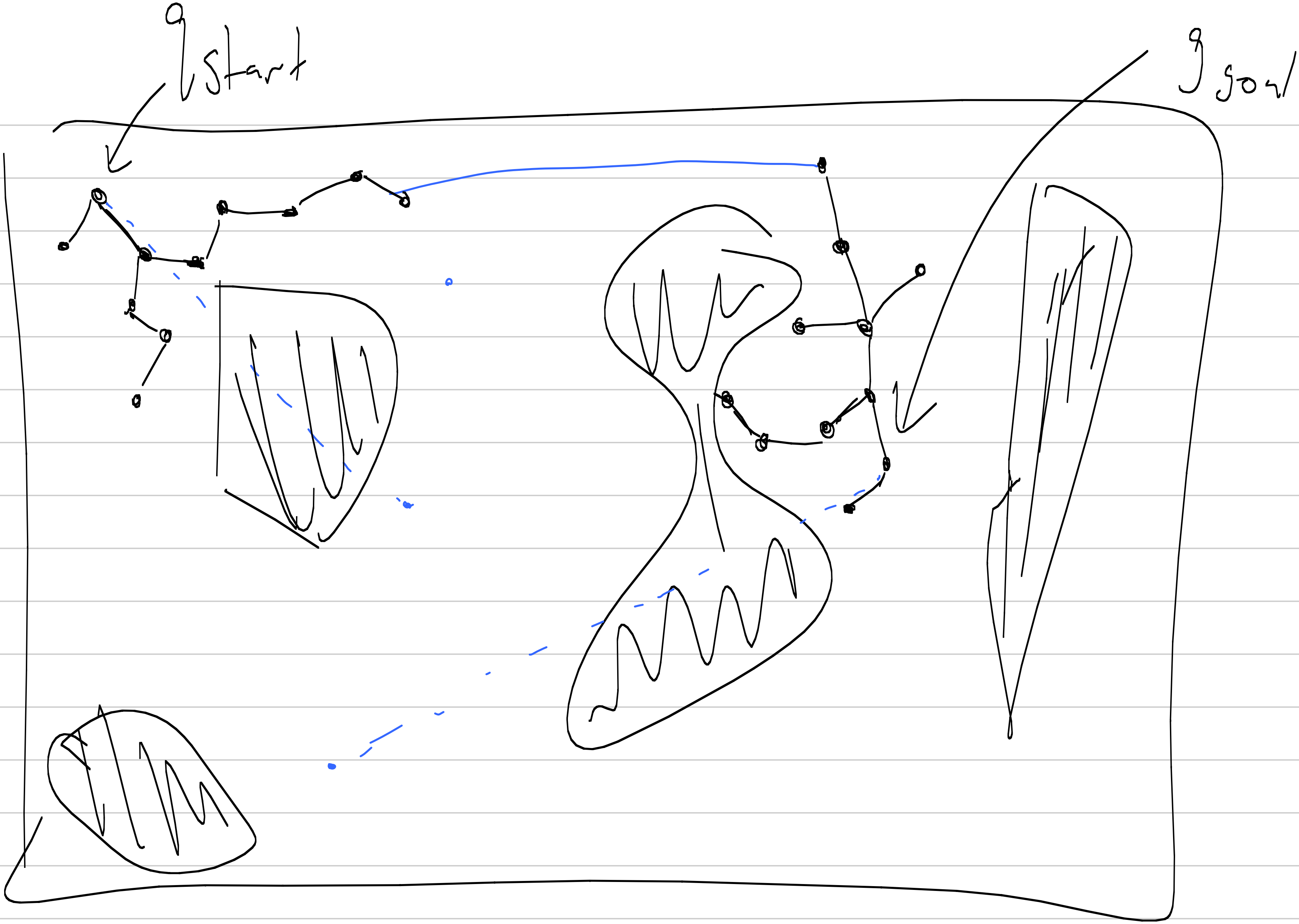
  if successful → Done!

## Sample

- Uniform
- $q_{rand} \sim N(q_{goal}, \Sigma)$
- $q_{rand} = q_{goal}$
- all of the above

## Step-size

- constant
- greedy (keep adding nodes until hitting an obstacle)
- dynamic, based on distance from goal



$q_{start}$
$q_{goal}$