Robotics I: Introduction to Robotics
Exercise 4 – Motion Planning

Jonas Kiemel, Tamim Asfour

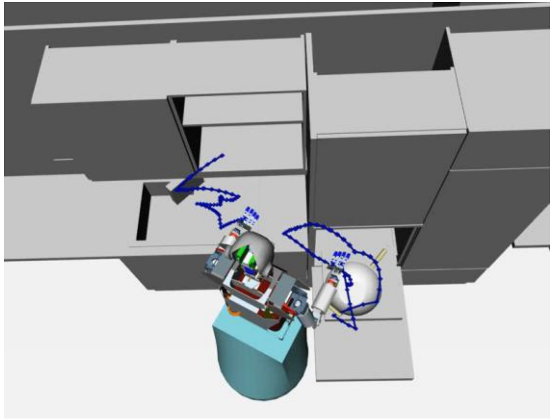
http://www.humanoids.kit.edu

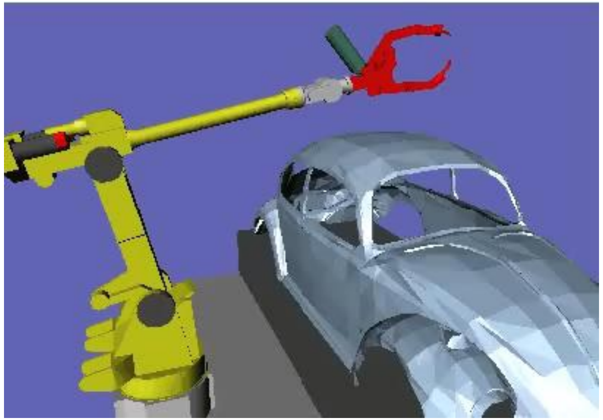
Exercises for Motion Planning

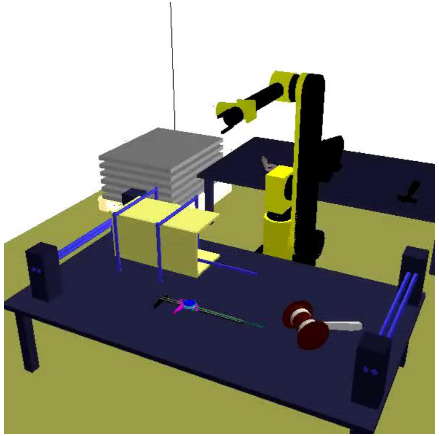
1. Voronoi diagram
2. Line Sweep
3. RRT*
4. A*
5. Potential Fields

Motion Planning: Motivation

Generation of a **collision-free** trajectory w.r.t. various **goals** and **constraints**







Motion Planning: Problem Statement

Given:

- Configuration space C
- Start configuration $\boldsymbol{q}_{start} \in C$
- Goal configuration $\boldsymbol{q}_{goal} \in C$

Required

Continuous trajectory

$$\tau: [0,1] \rightarrow C \text{ with } \tau(0) = \boldsymbol{q}_{start} \text{ and } \tau(1) = \boldsymbol{q}_{goal}$$

With respect to

- Kinematic constraints (joint limits, maximal acceleration, …)
- Quality criteria (duration, energy, distance to obstacles, smoothness of the trajectory, …)
- Additional contraints (upright position of the end-effector, …)

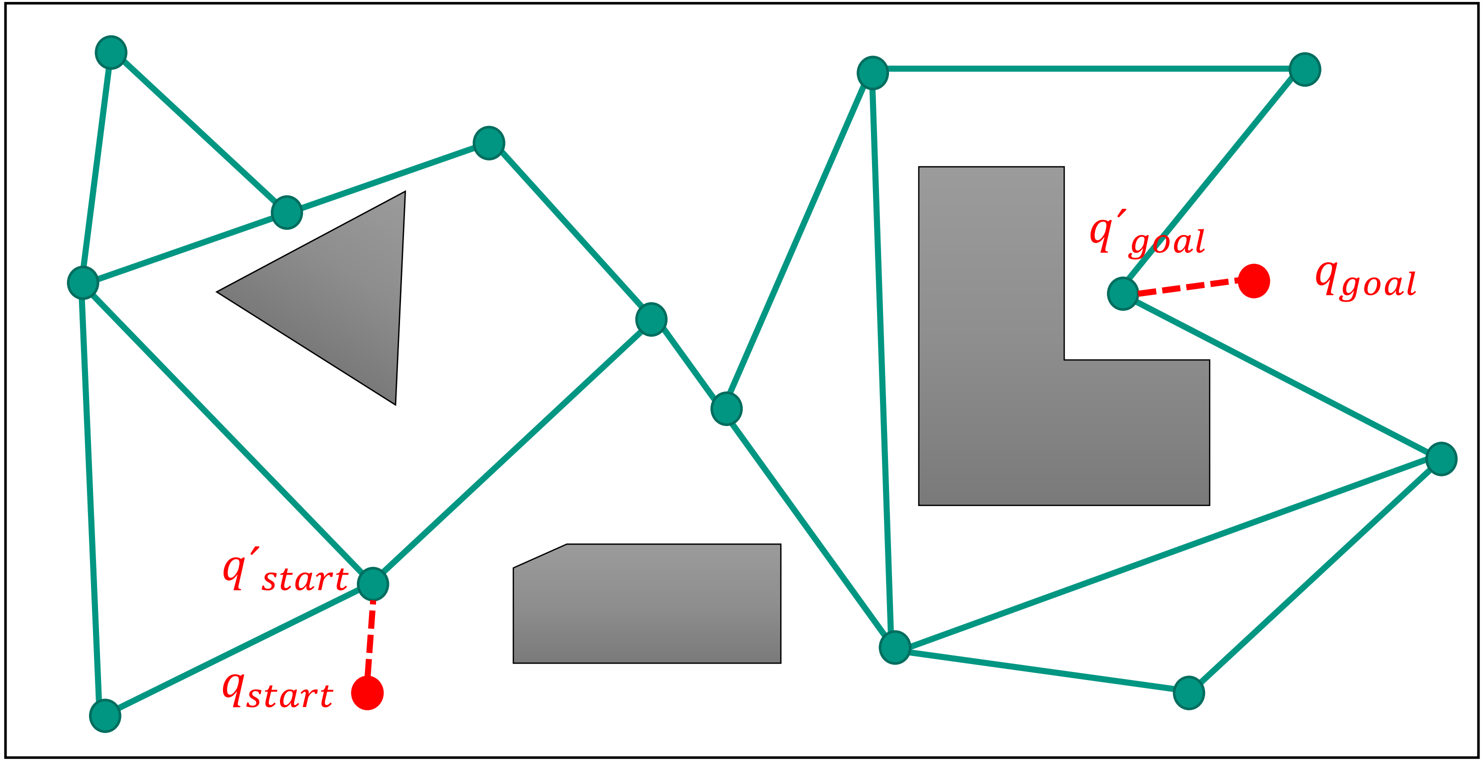
Motion Planning for Mobile Robots: Graphs

- Calculating collision-free trajectories

  - 1. Step: Efficient representation of free space by network of paths (graph)
    - Voronoi diagram
    - Cell decomposition (e.g. using **Line-Sweep**)

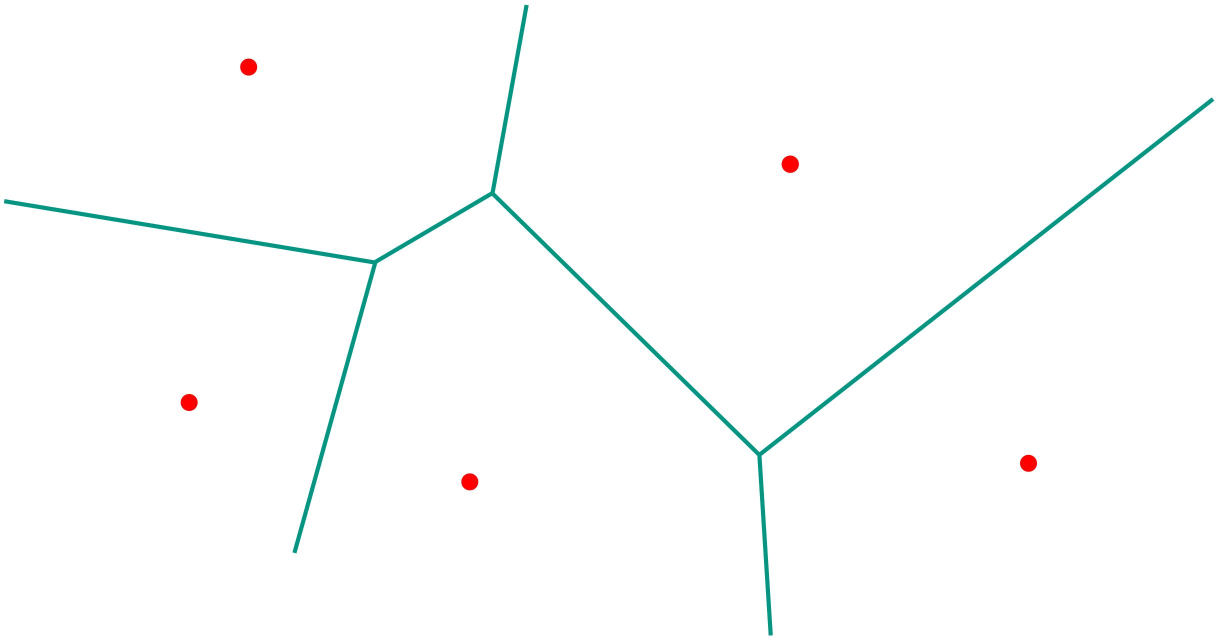
  - 2. Step: Search optimal path in graph
    - A*

Motion Planning for Mobile Robots: Graphs



Exercise 1: Voronoi Diagrams

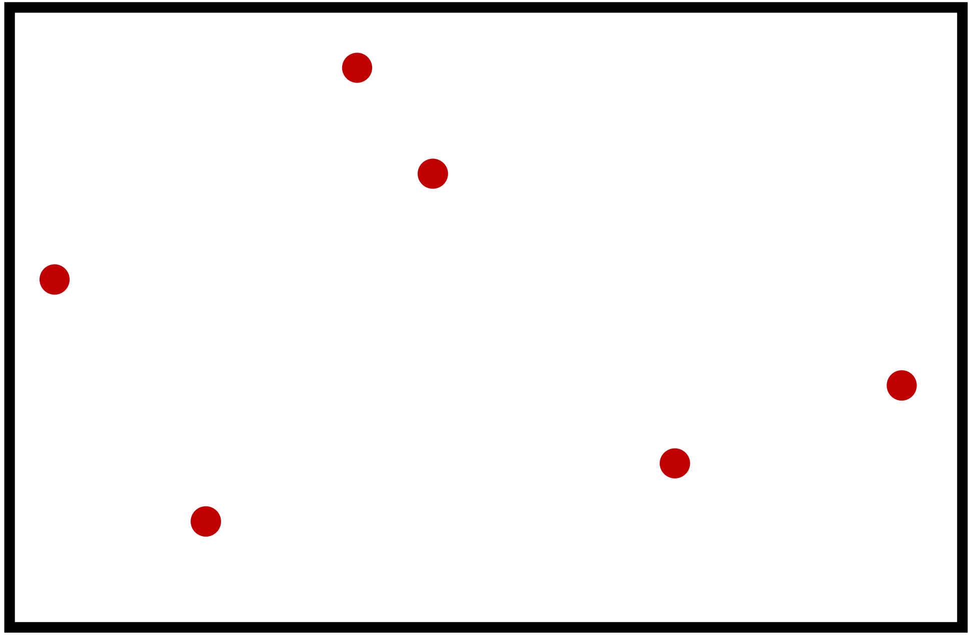
- Voronoi diagram for **P**



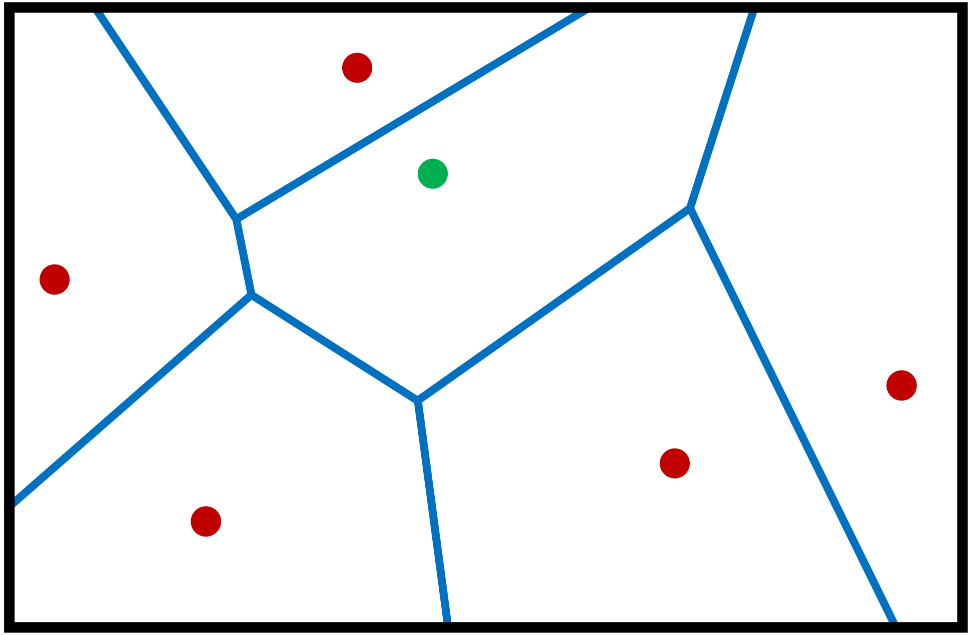
Exercise 1: Voronoi Diagrams

1. Explain the terms
   - Voronoi region
   - Voronoi edge
   - Voronoi vertex

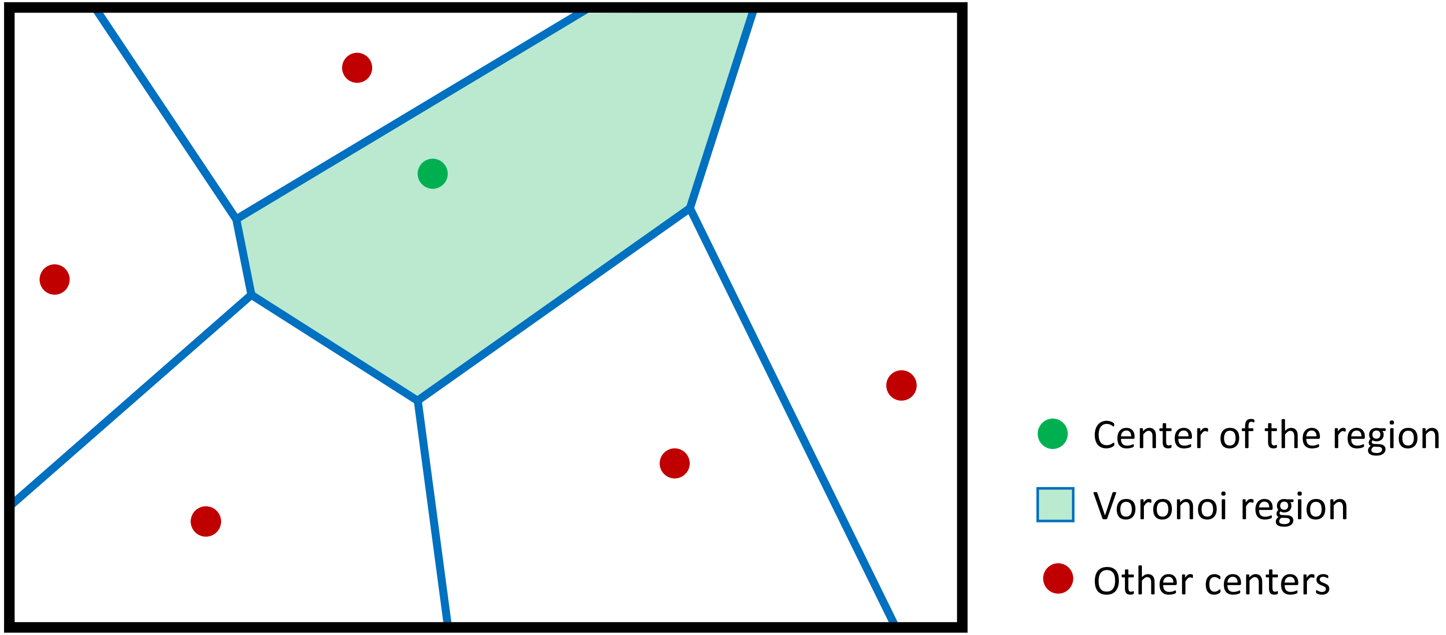
2. Find the Voronoi diagram for the point set P:



Exercise 1.1: Voronoi Terms, Region (1)

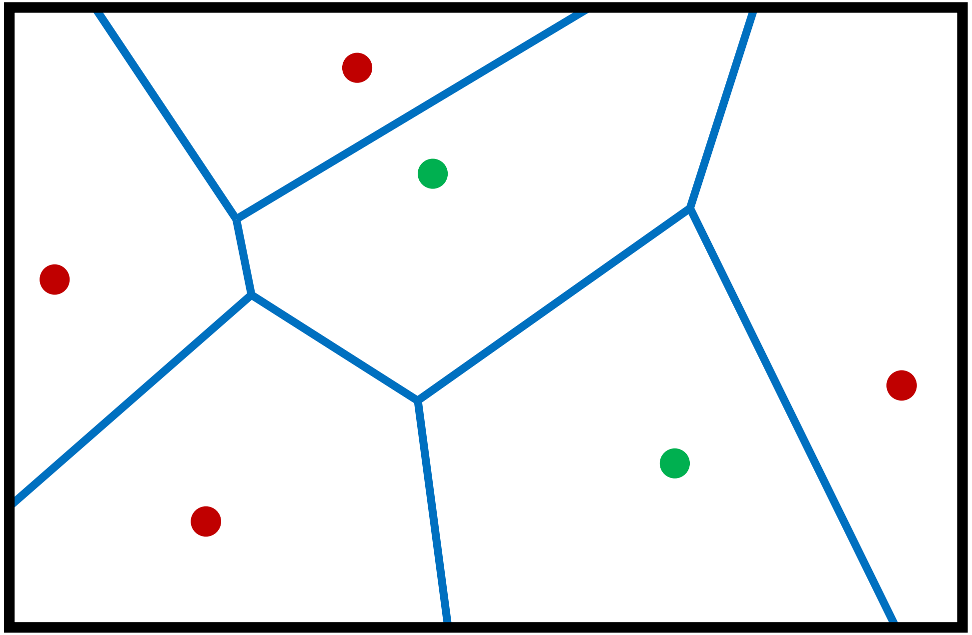


Exercise 1.1: Voronoi Terms, Region (2)

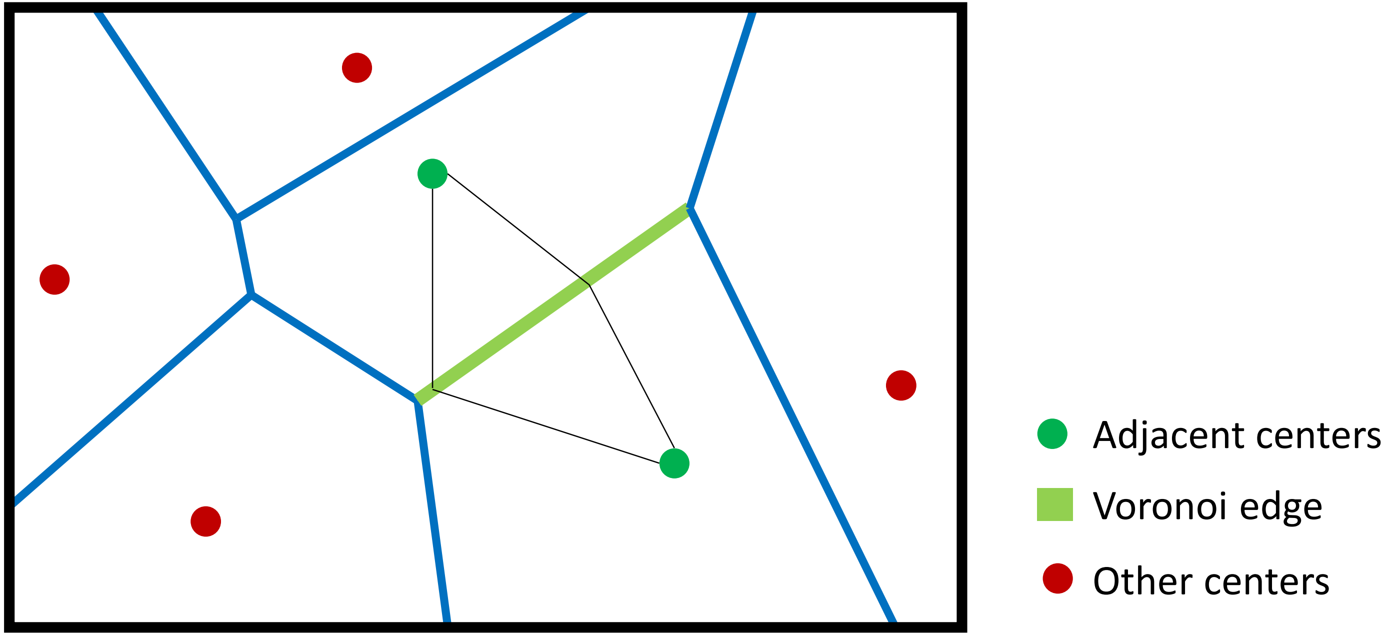


- Voronoi region:
  A **region** is defined as the **set of points** whose **distance** to a **center** is **less than the distance to all other centers**.

Exercise 1.1: Voronoi Terms, Edge (1)

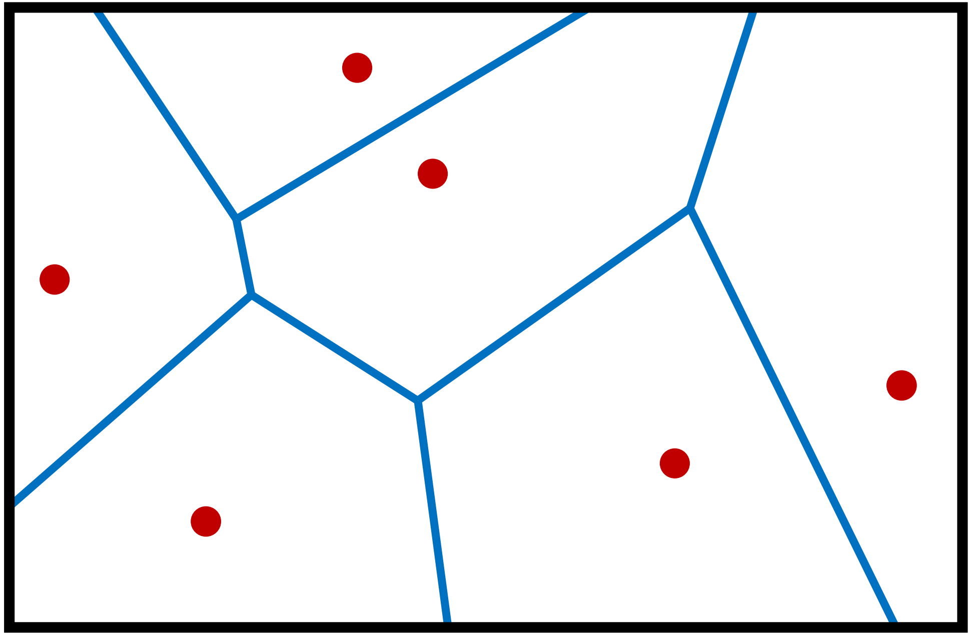


Exercise 1.1: Voronoi Terms, Edge (2)

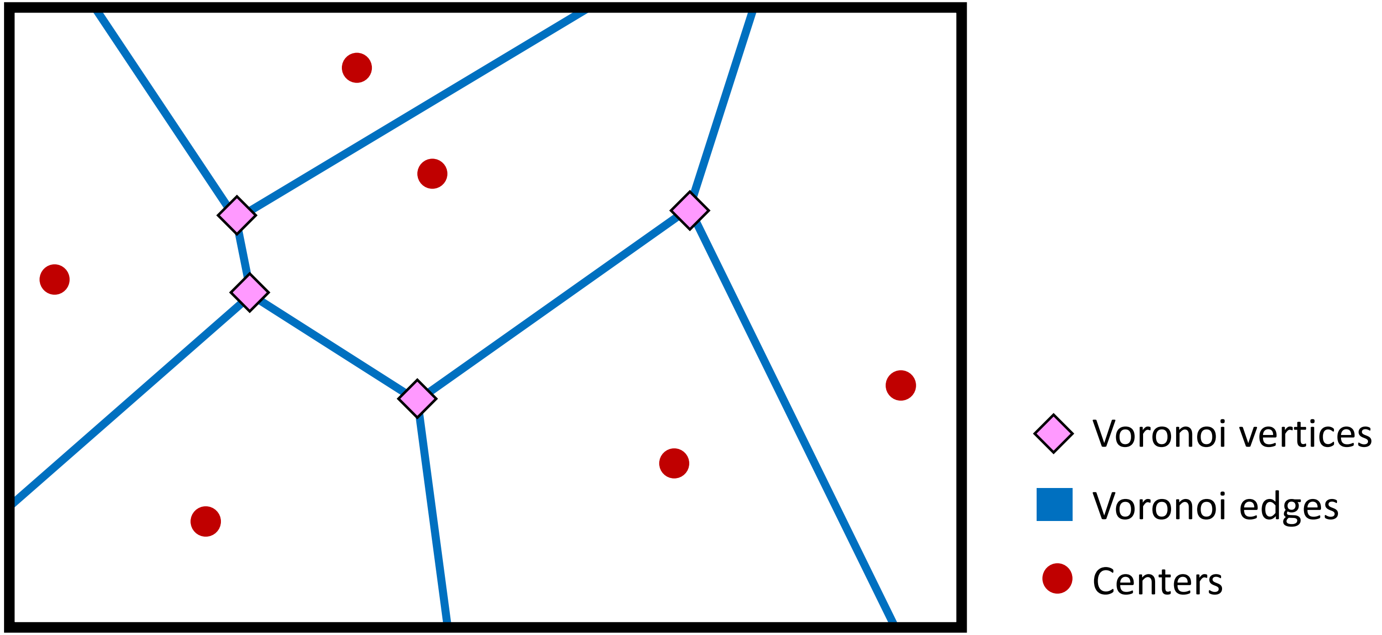


- Voronoi edge:
  All **points of a Voronoi edge** have the **same distance** to the centers of **the adjacent regions.**

Exercise 1.1: Voronoi Terms, Vertices (1)

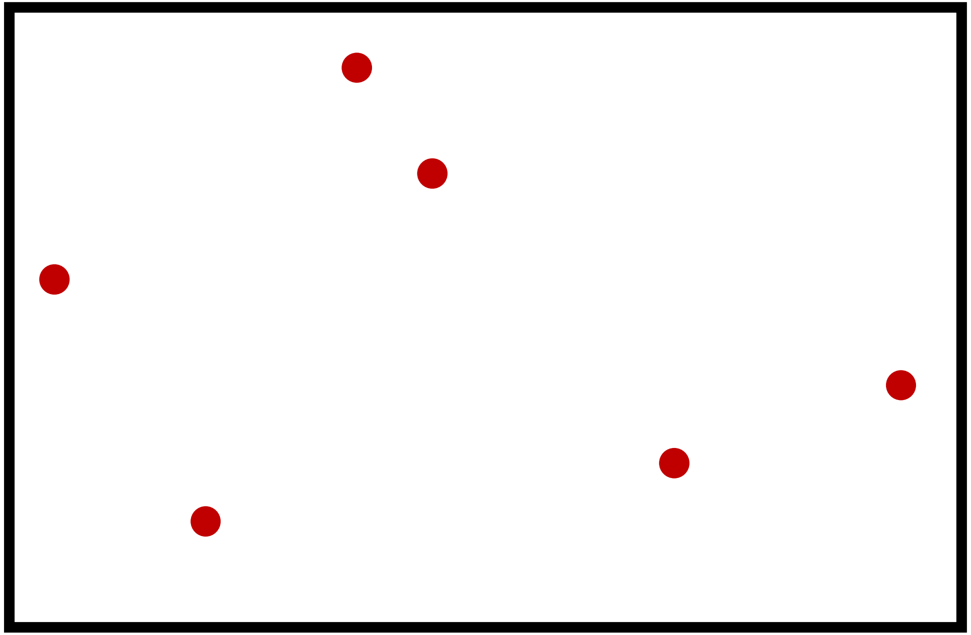


Exercise 1.1: Voronoi Terms, Vertices (2)

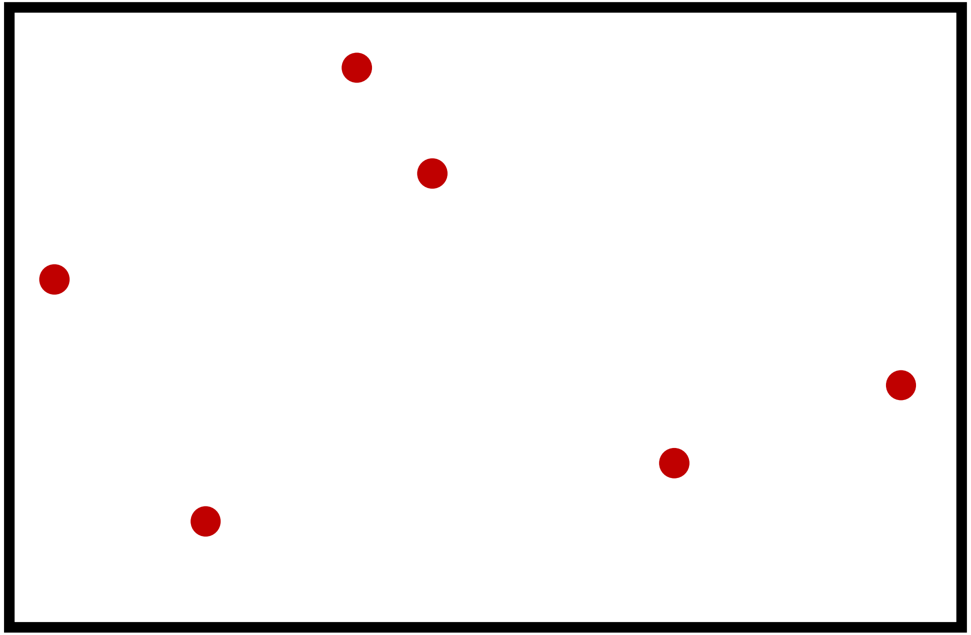


- Voronoi vertices:
  Corners of the polygons/voronoi regions

Exercise 1.2: Voronoi diagram for P (1)

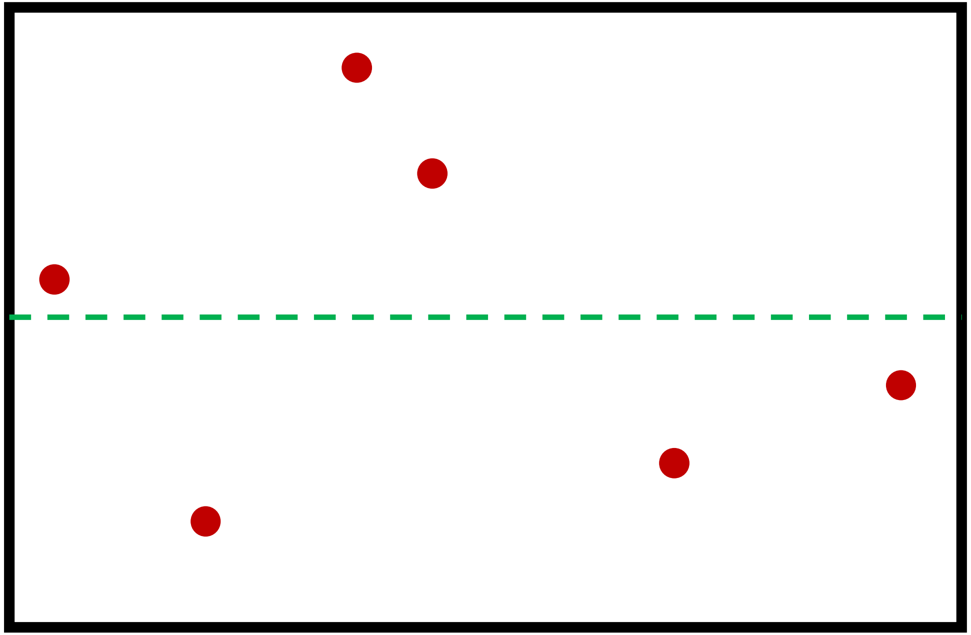


Exercise 1.2: Voronoi diagram for P (2)



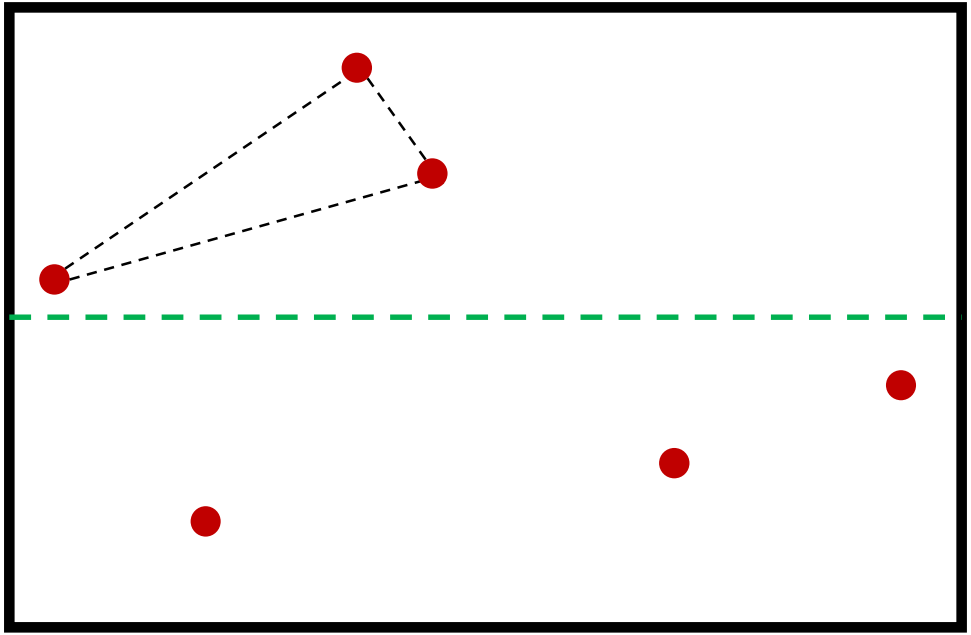
1. Recursively split the set of points in half

Exercise 1.2: Voronoi diagram for P (3)



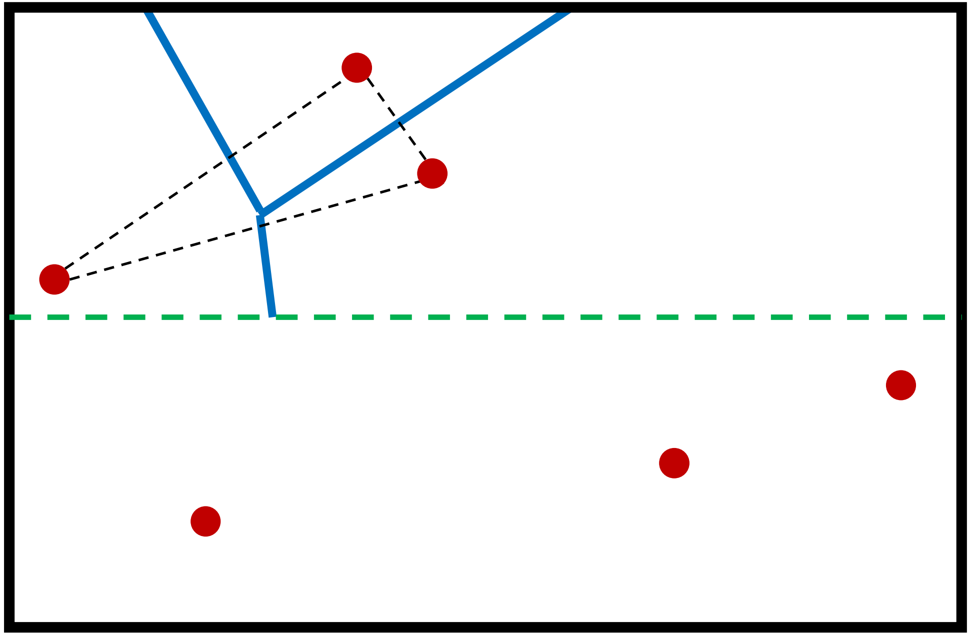
1. Recursively split the set of points in half
2. Solve the base case

Exercise 1.2: Voronoi diagram for P (4)



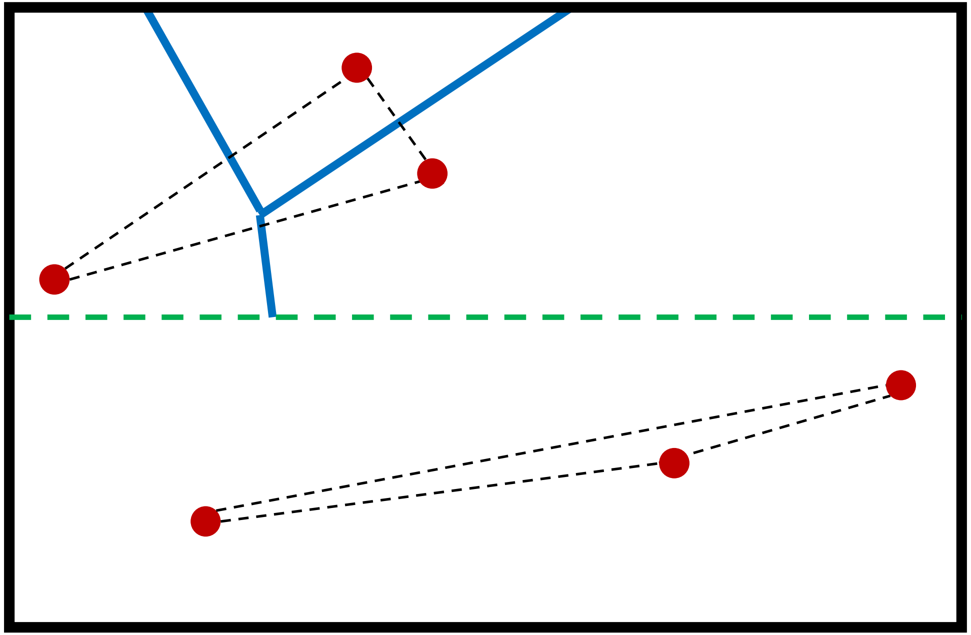
1. Recursively split the set of points in half
2. Solve the base case

Exercise 1.2: Voronoi diagram for P (5)



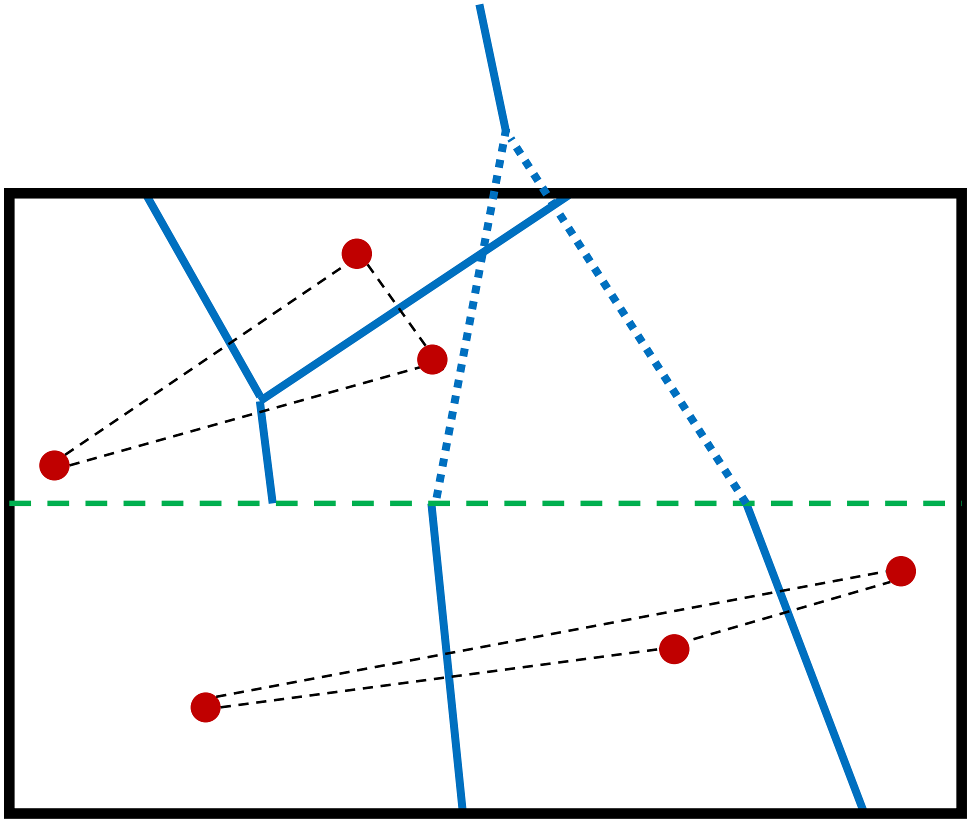
1. Recursively split the set of points in half
2. Solve the base case
   - Perpendicular bisector

Exercise 1.2: Voronoi diagram for P (6)



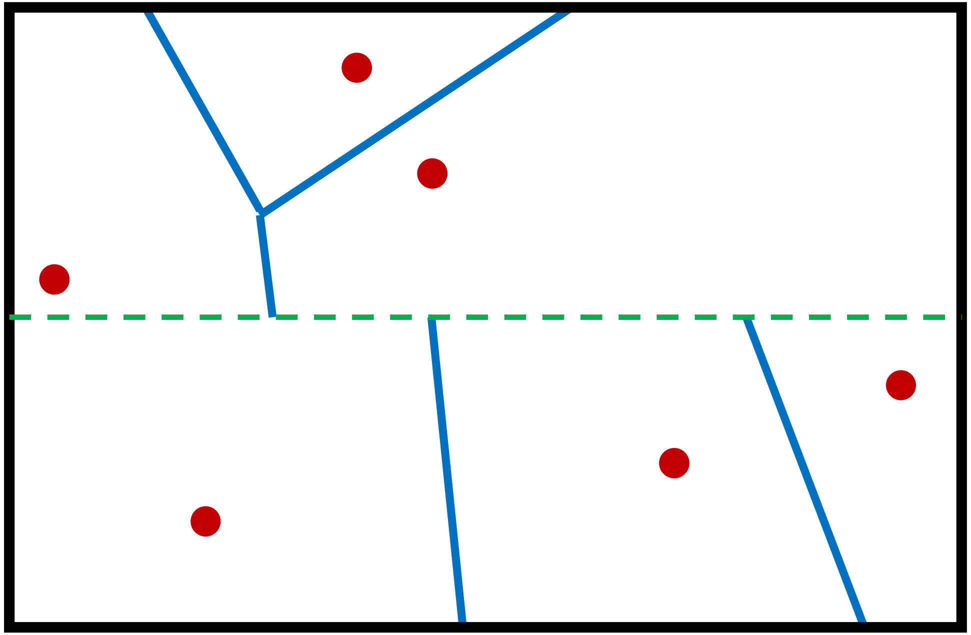
1. Recursively split the set of points in half
2. Solve the base case
   - Perpendicular bisector

Aufgabe 1.2: Voronoi-Diagramm für P (7)



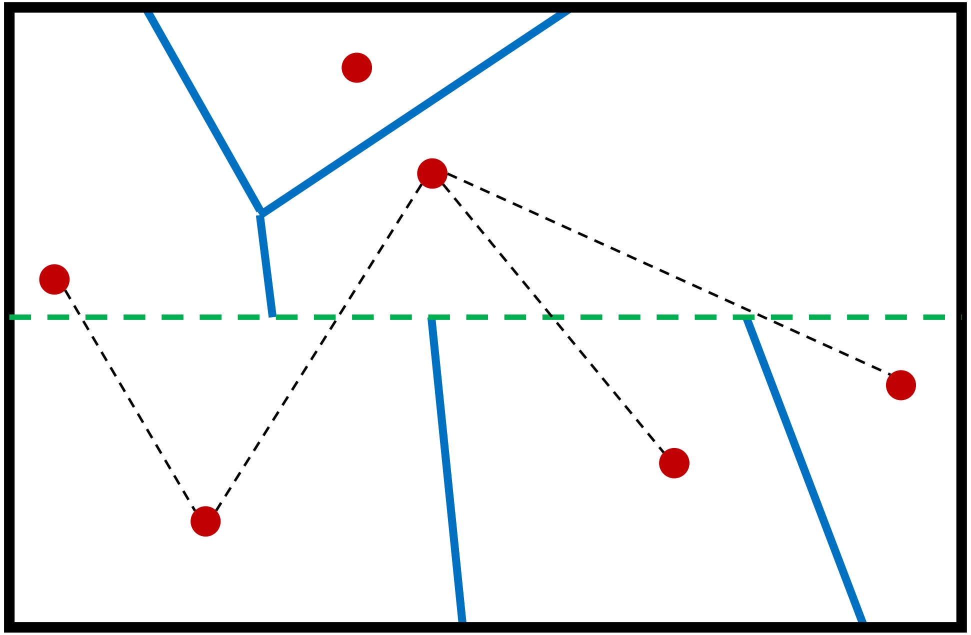
1. Recursively split the set of points in half
2. Solve the base case
   - Perpendicular bisector

Exercise 1.2: Voronoi diagram for P (8)



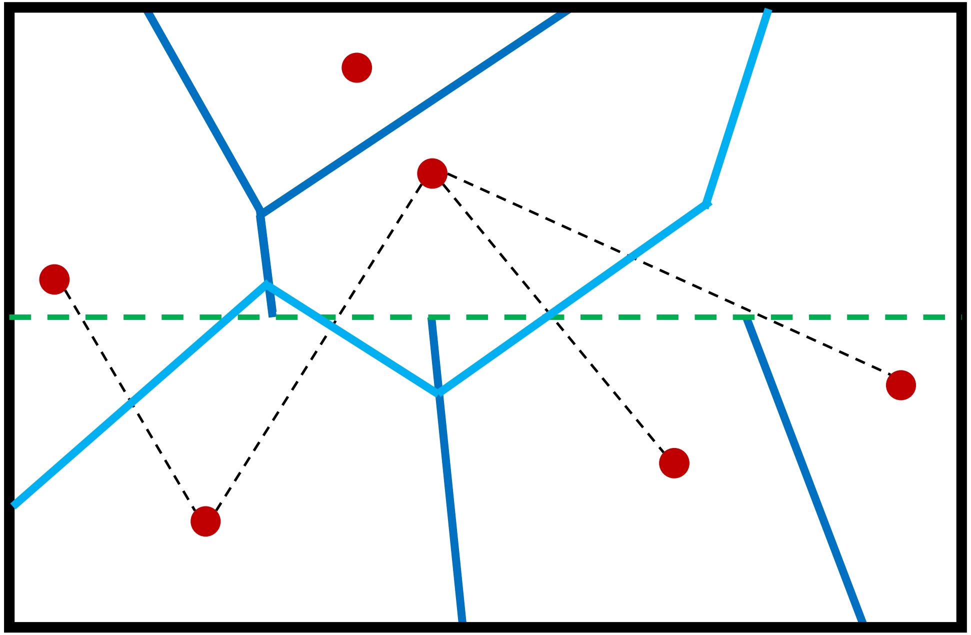
1. Recursively split the set of points in half
2. Solve the base case
   - Perpendicular bisector
3. Connect neighbors

Exercise 1.2: Voronoi diagram for P (9)



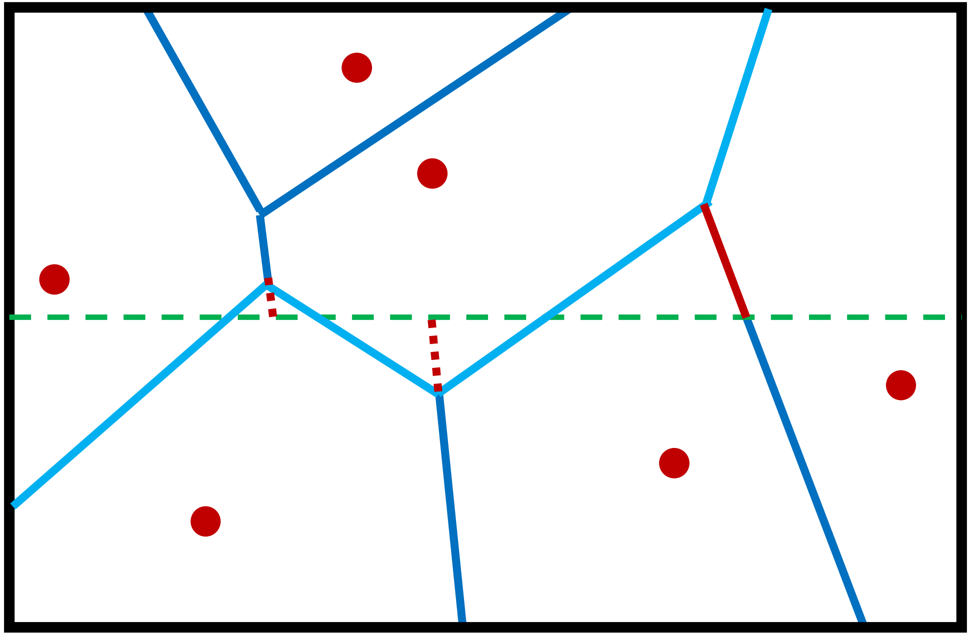
1. Recursively split the set of points in half
2. Solve the base case
   - Perpendicular bisector
3. Connect neighbors
   - Perpendicular bisector

Exercise 1.2: Voronoi diagram for P (10)



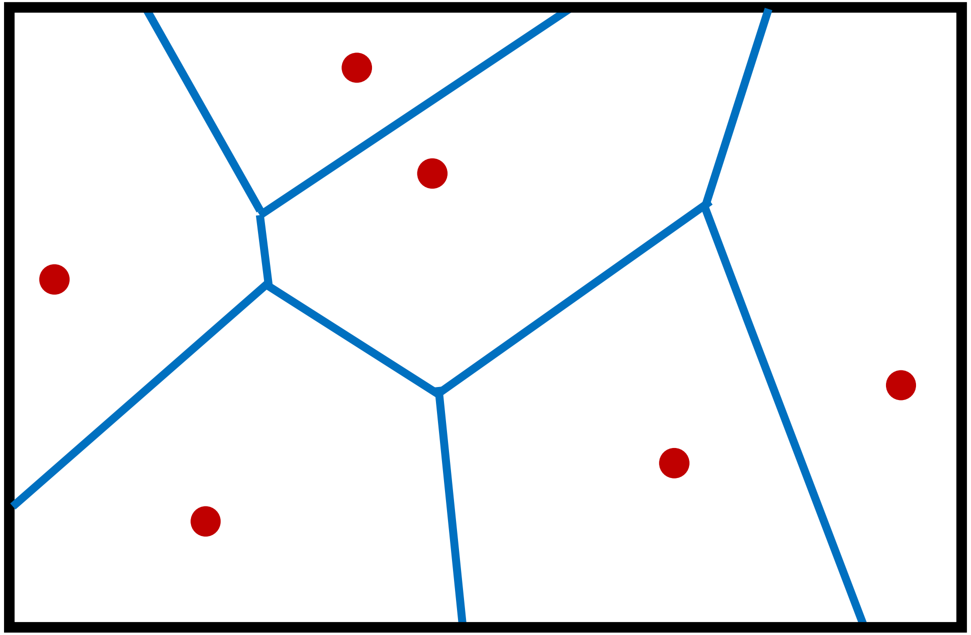
1. Recursively split the set of points in half
2. Solve the base case
   - Perpendicular bisector
3. Connect neighbors
   - Perpendicular bisector
4. Close regions
   - Shorten or extend lines

Exercise 1.2: Voronoi diagram for P (11)



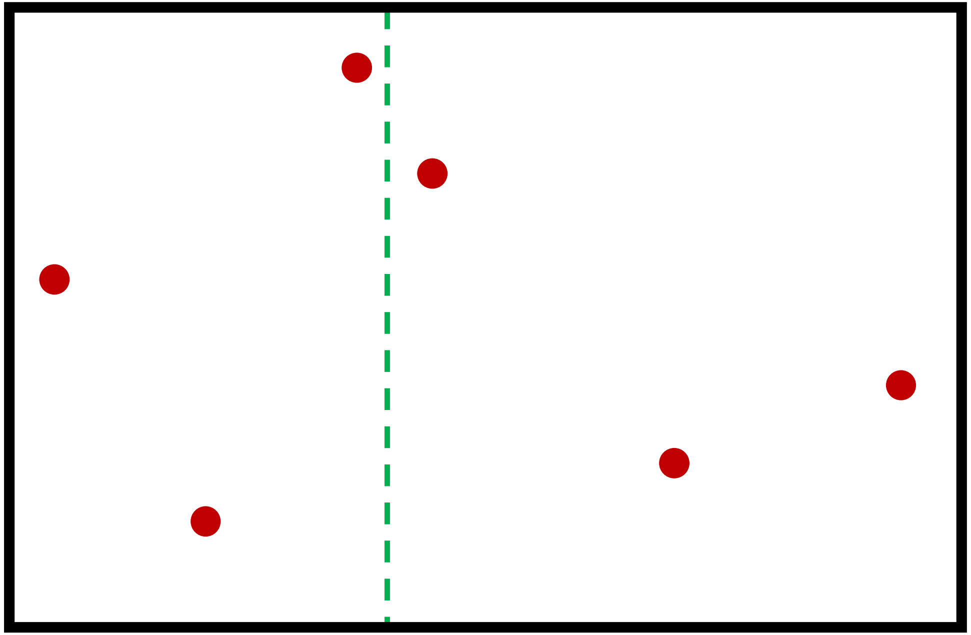
1. Recursively split the set of points in half
2. Solve the base case
   - Perpendicular bisector
3. Connect neighbors
   - Perpendicular bisector
4. Close regions
   - Shorten or extend lines

Exercise 1.2: Voronoi diagram for P (12)



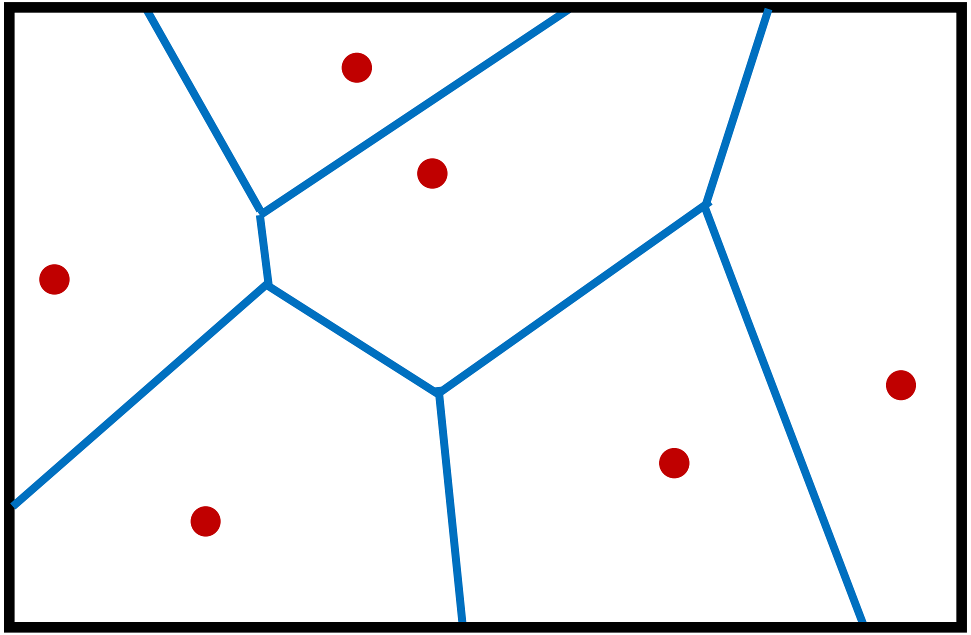
1. Recursively split the set of points in half
2. Solve the base case
   - Perpendicular bisector
3. Connect neighbors
   - Perpendicular bisector
4. Close regions
   - Shorten or extend lines

Exercise 1: Voronoi diagram for P, Bonus



Which diagram results if the point set is divided differently?

Exercise 1: Voronoi diagram for P, Bonus

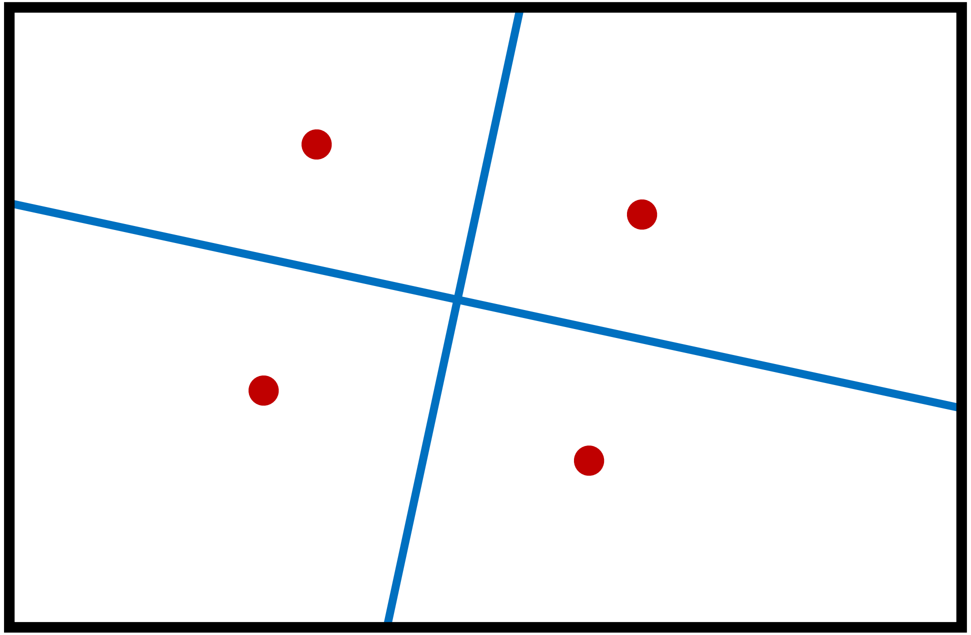


Which diagram results if the point set is divided differently?

Exercise 1: Voronoi diagram for P, Bonus

Can there be Voronoi vertices where more than three Voronoi edges come together?

Exercise 1: Voronoi diagram for P, Bonus

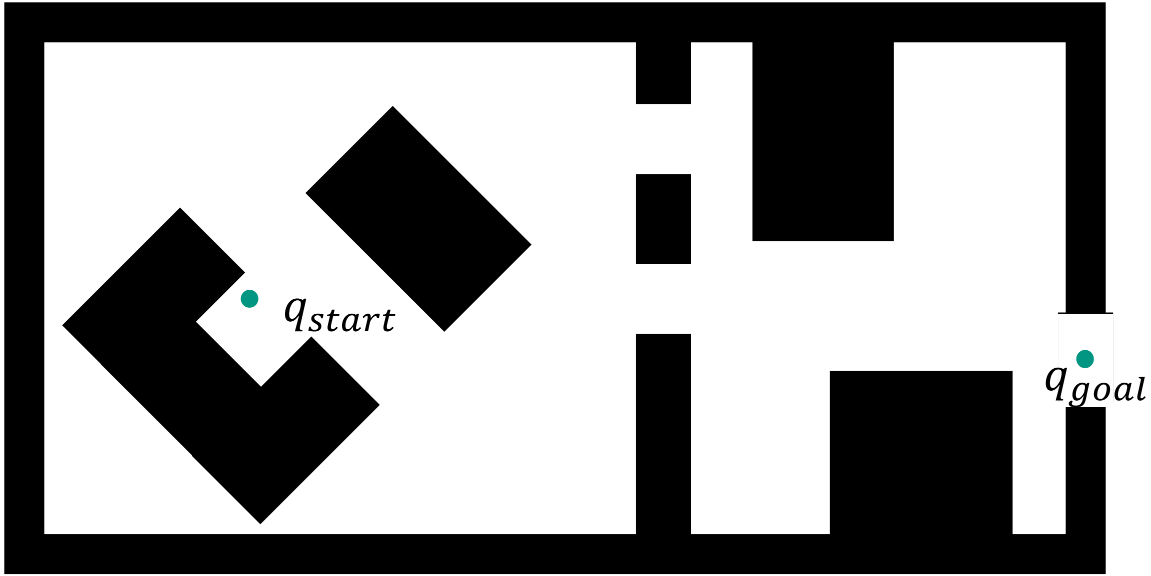


Can there be Voronoi vertices where more than three Voronoi edges come together?

Cell Decomposition

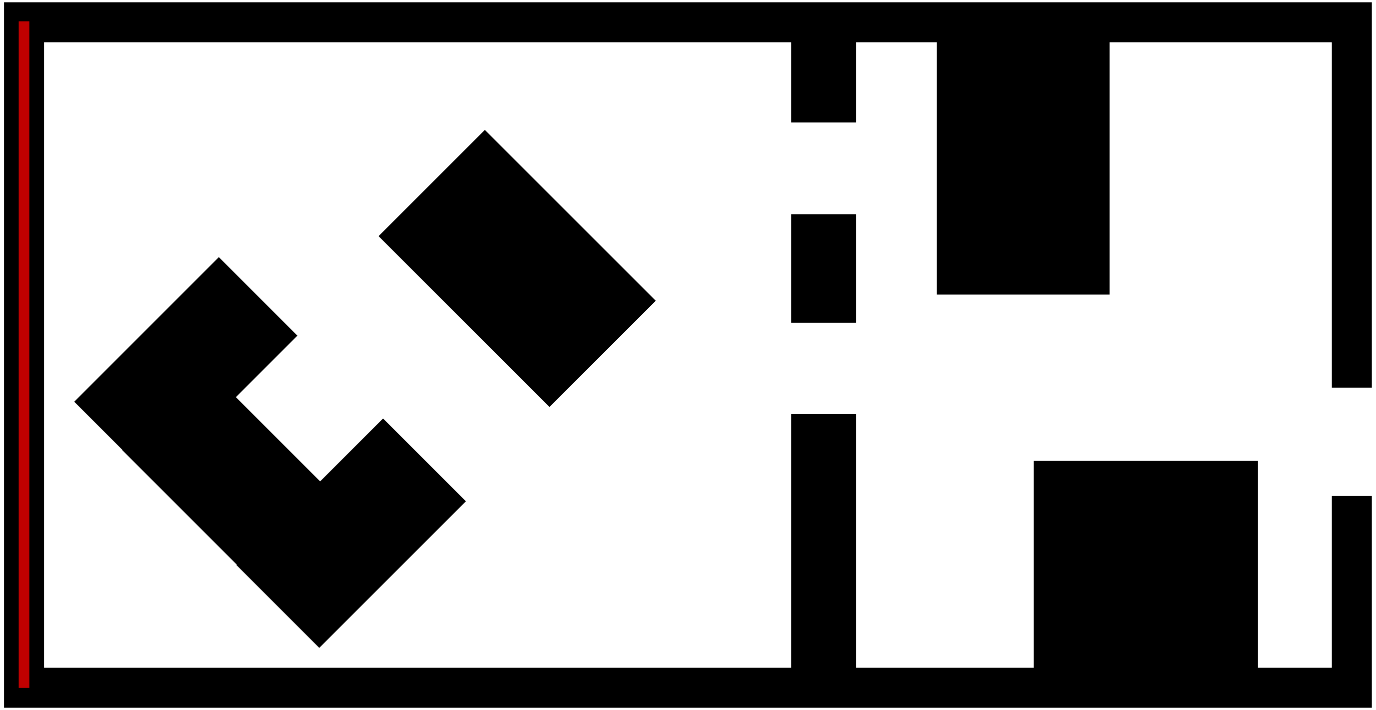
- Approach:
  1. decompose C_{free} in cells, that make it easy to find a path between two configurations within the cell
  2. Represent the spatial layout by an adjacency graph
  3. Search the optimal path from $\boldsymbol{q}_{start}$ to $\boldsymbol{q}_{goal}$ in the graph
- There are two kinds of cell decomposition:
  - Exact cell decomposition (e.g. using **Line-Sweep**)
  - Approximated cell decomposition

Exercise 2: Line-Sweep

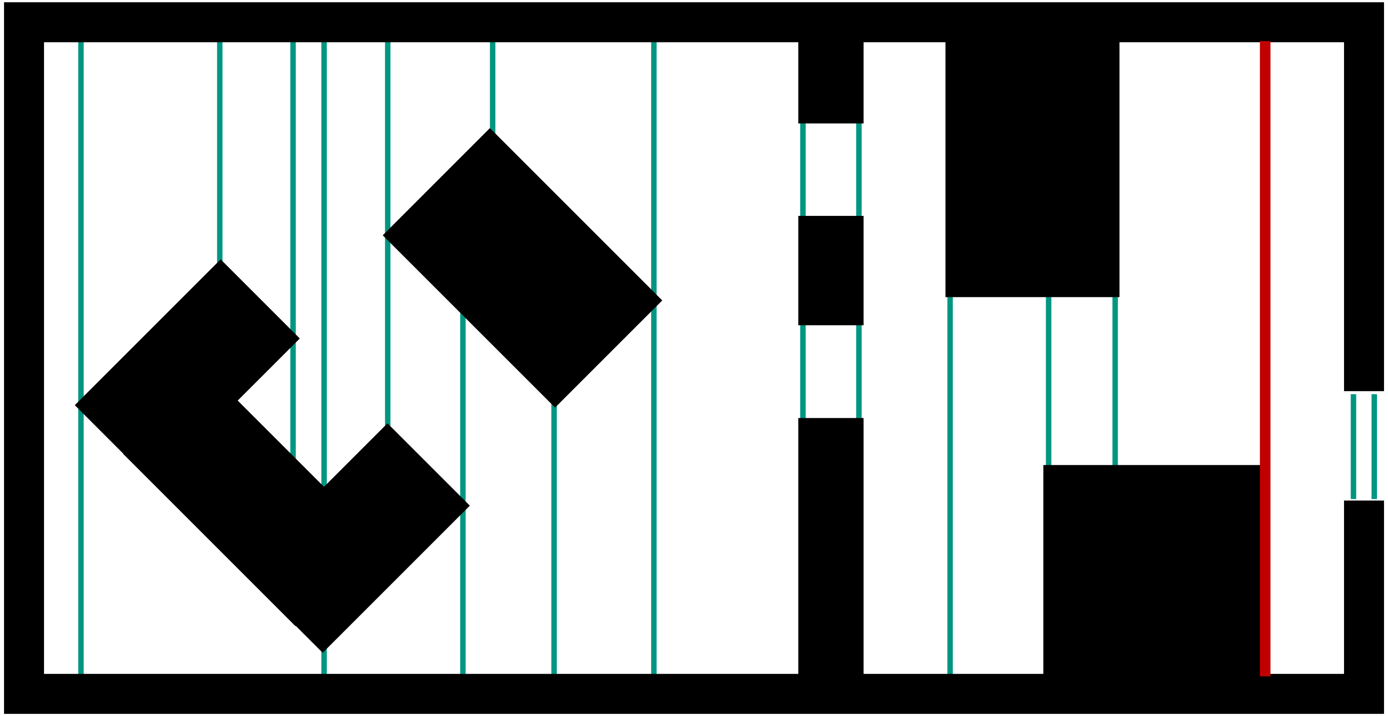


1. Cell decomposition using Line-Sweep
2. Adjacency graph of the cells
3. Path between q_{start} and q_{goal}

Exercise 2.1: Line-Sweep, Cell decomposition (1)

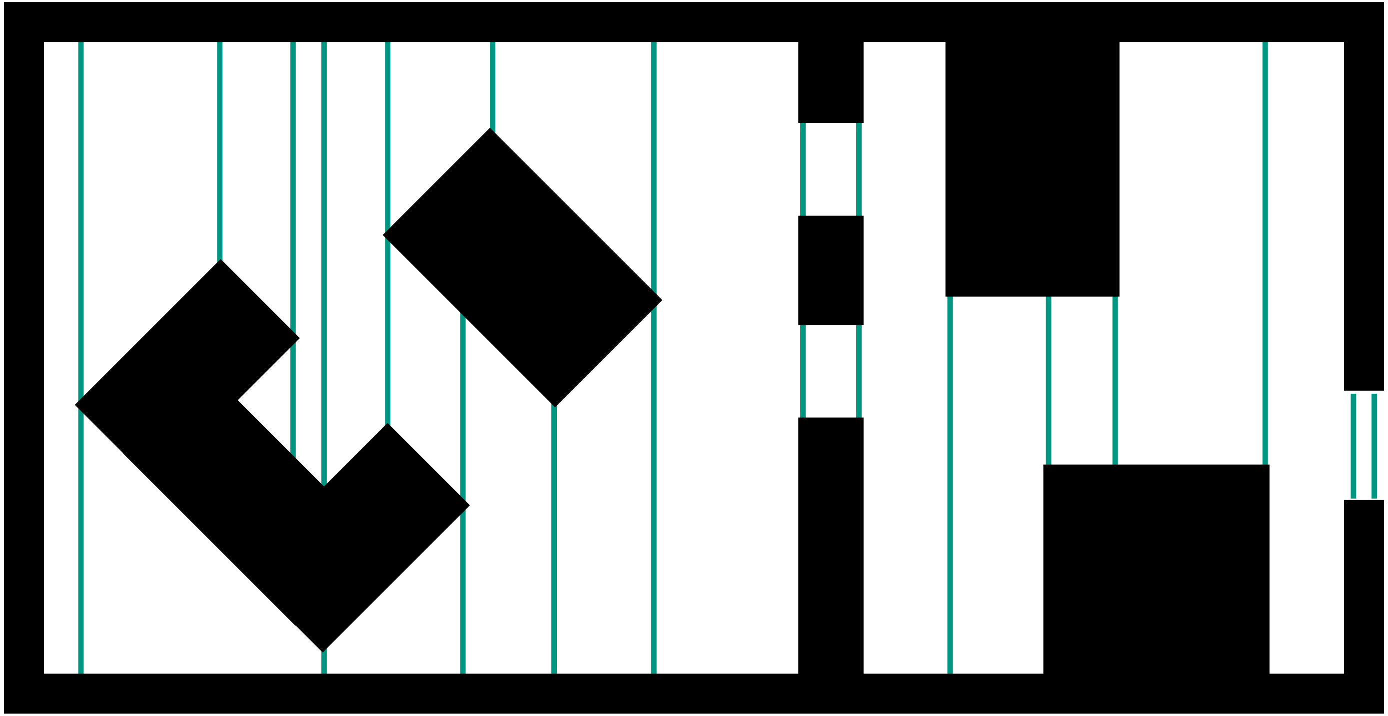


Exercise 2.1: Line-Sweep, Cell decomposition (2)



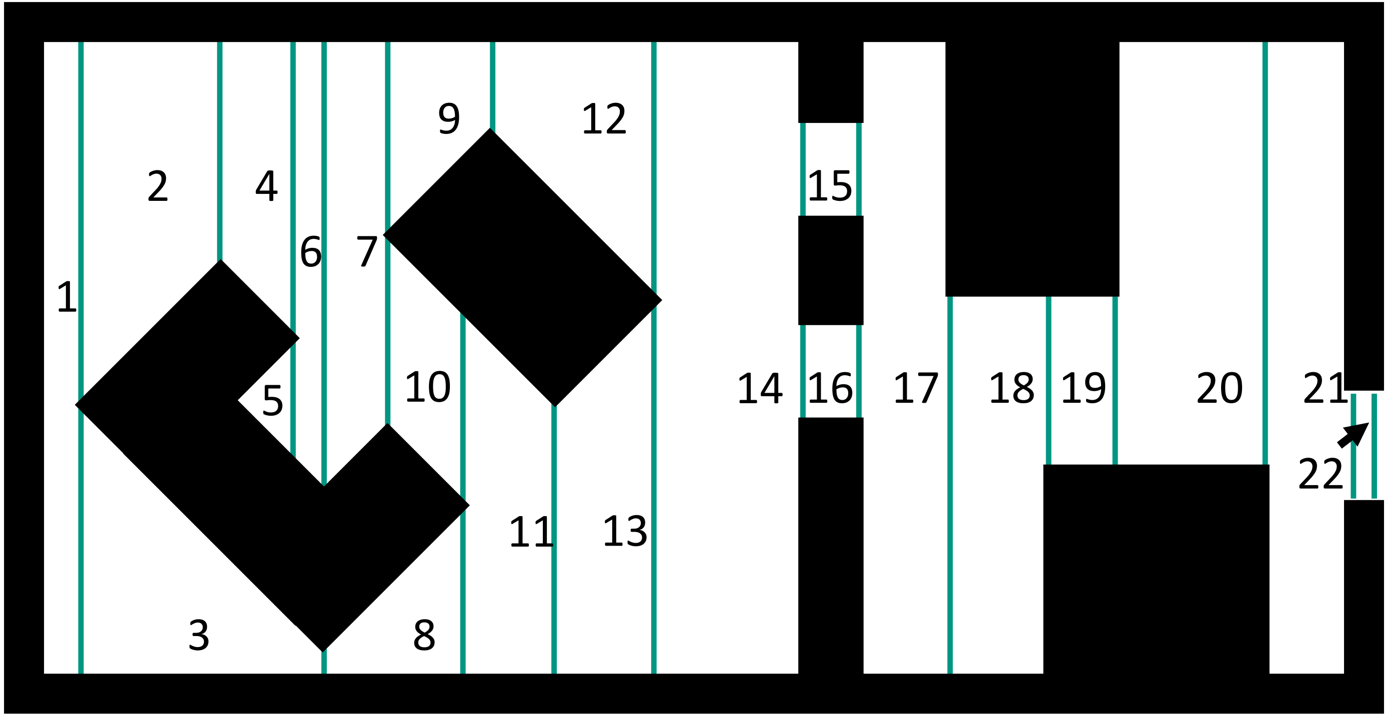
Exercise 2.1: Line-Sweep, Cell decomposition (1)

Numbering from **left to right**, then from **top to bottom**

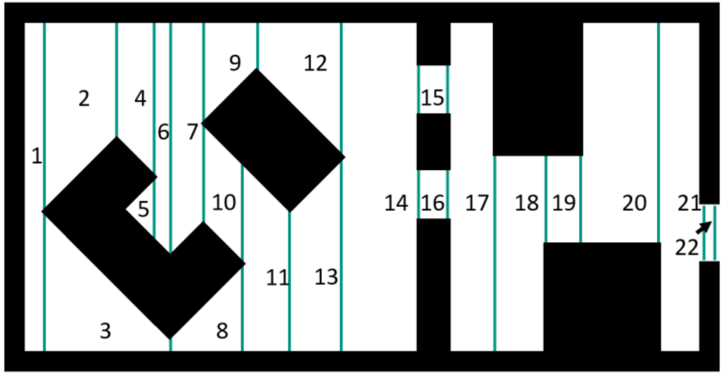


Exercise 2.1: Line-Sweep, Cell decomposition (2)

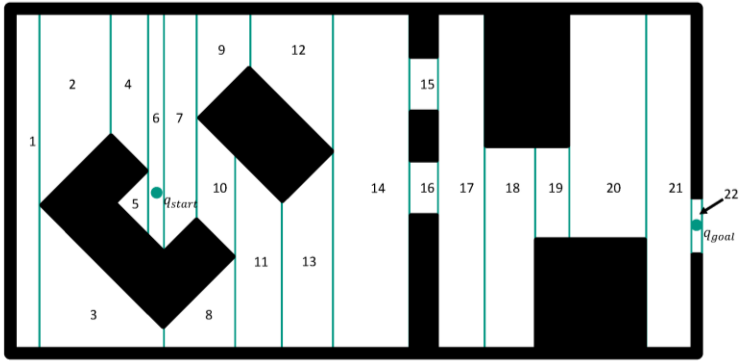
Numbering from **left to right**, then from **top to bottom**

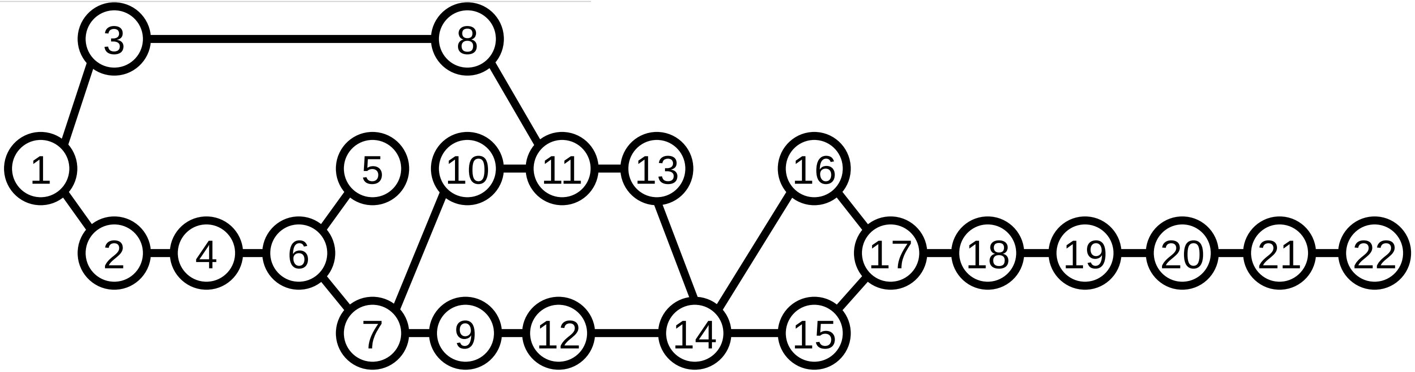


Exercise 2.2: Line-Sweep, Adjacency graph (1)

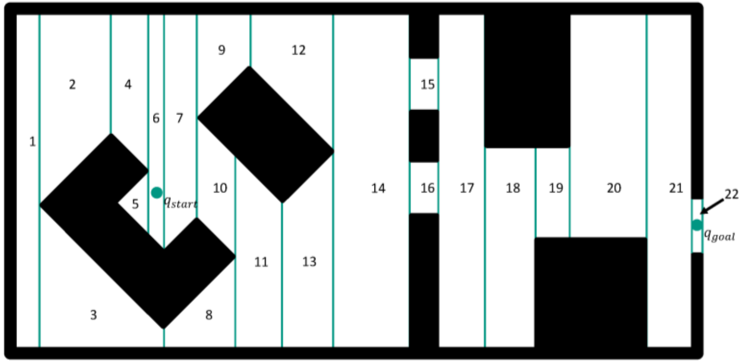


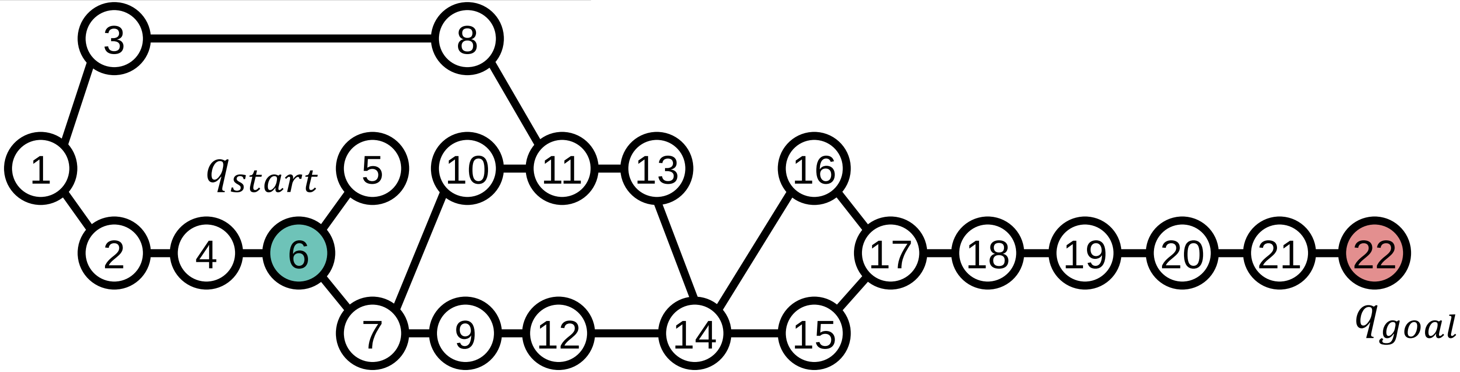
Exercise 2.2: Line-Sweep, Adjacency graph (2)



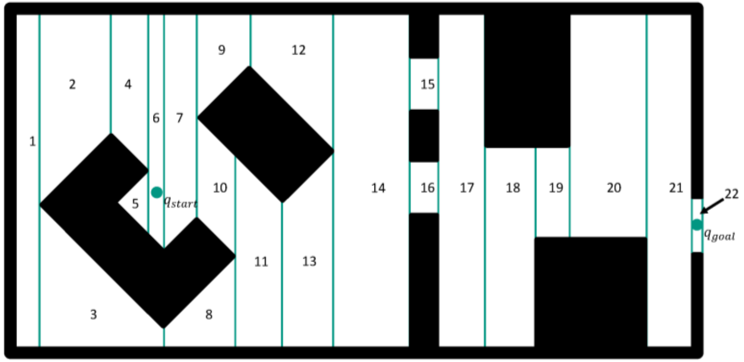


Exercise 2.3: Line-Sweep, Path from start to goal (1)



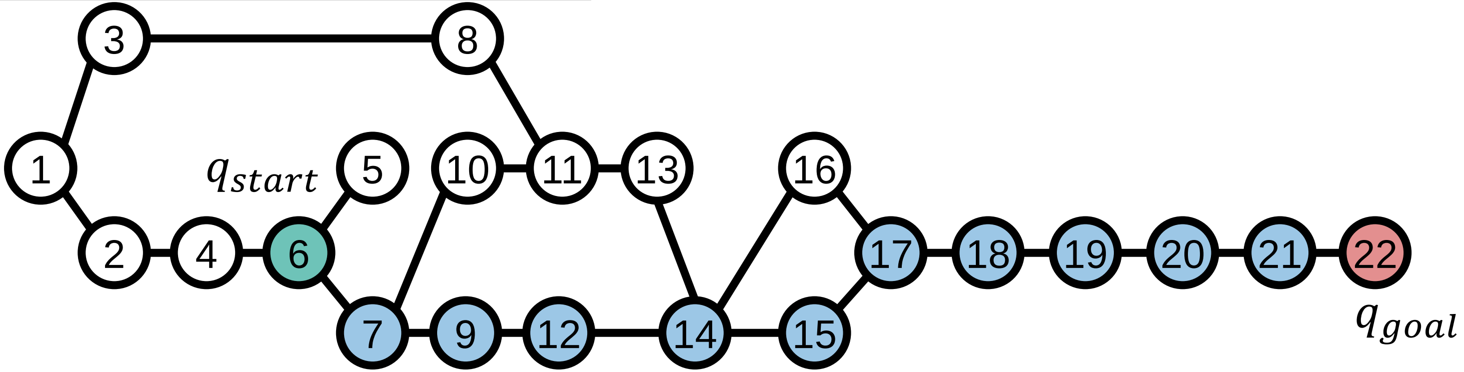


Exercise 2.3: Line-Sweep, Path from start to goal (2)



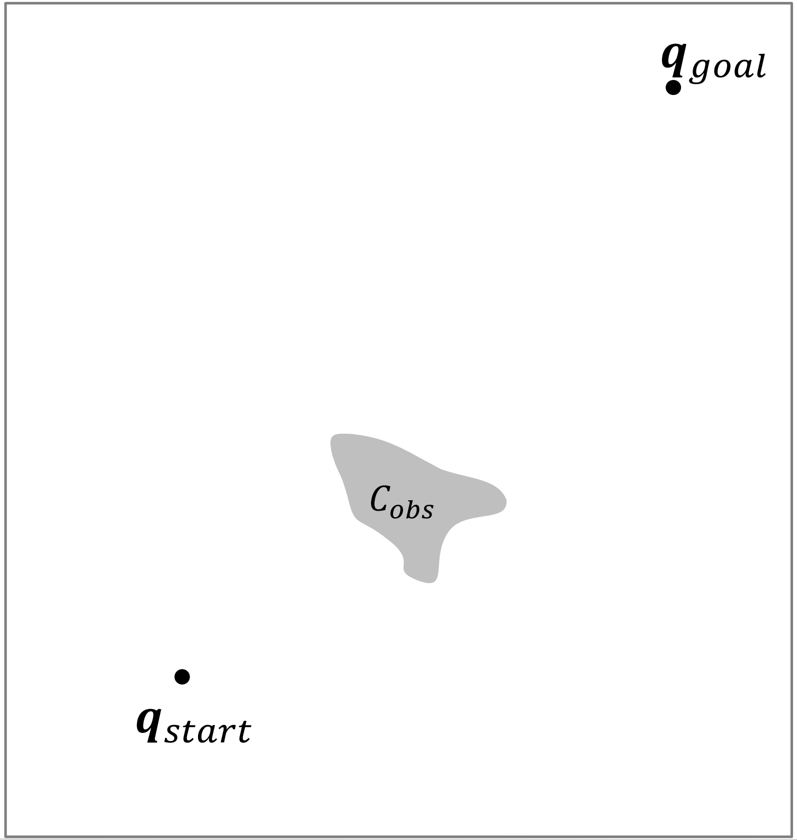
Path from q_{start} to q_{goal}:

6, 7, 9, 12, 14, 15, 17, 18, 19, 20, 21, 22



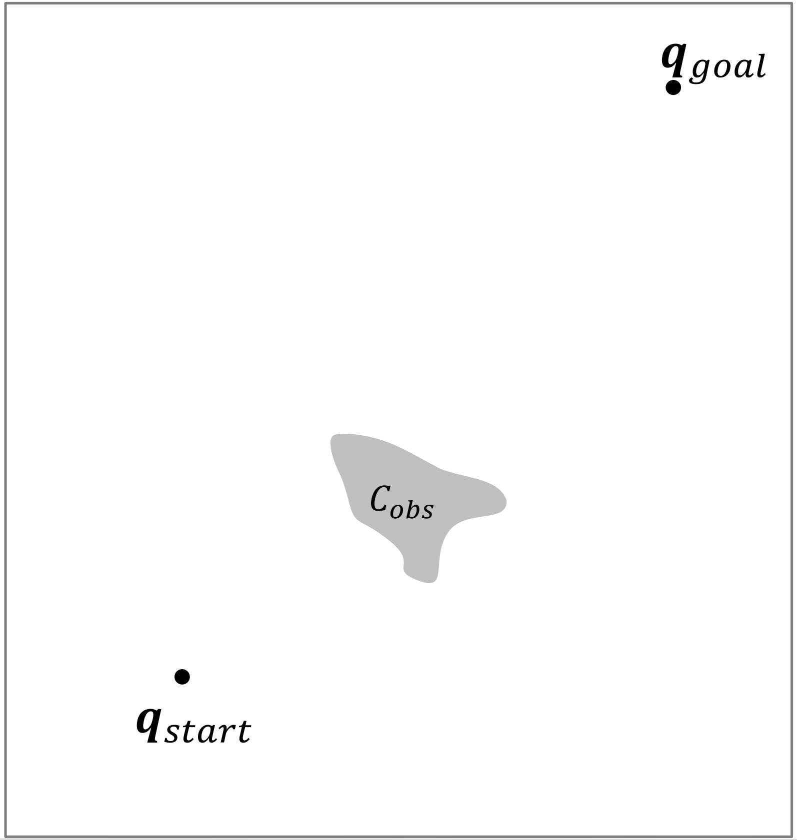
RRT: Principle (1)

- The shape of C_{obs} in the configuration space is **unknown**
- Initialization of the RRT
  - Create empty tree T
  - Insert $\boldsymbol{q}_{start}$ into T



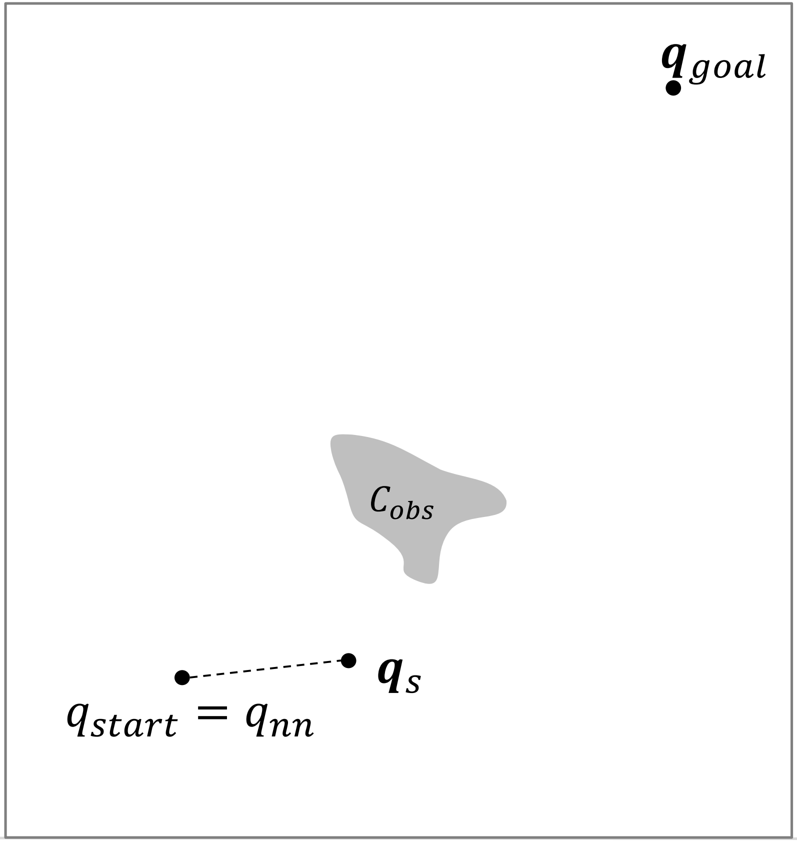
RRT: Principle (2)

- Iteration
  1. Sample a point $\boldsymbol{q}_s$
  2. Determine the next neighbor $\boldsymbol{q}_{nn}$ in T
  3. Add points on the connection between $\boldsymbol{q}_s$ and $\boldsymbol{q}_{nn}$ to T
     - With step size d
     - Check every part of the path for collision with C_{obs}
     - Stop when a collision has been detected
  4. Go to 1.



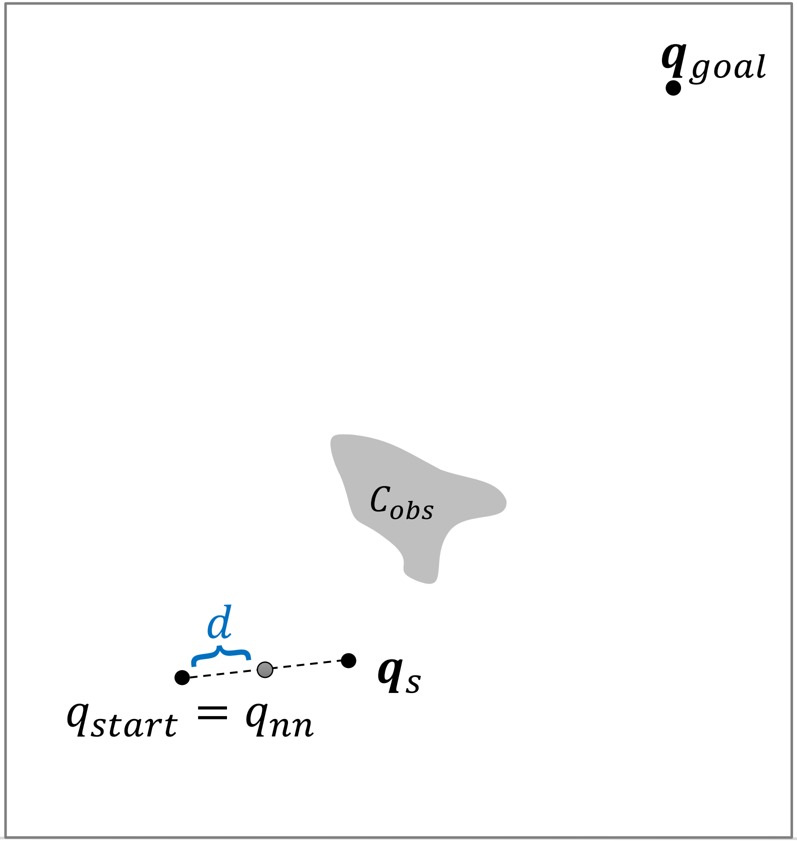
RRT: Principle (3)

- Iteration
  1. Sample a point $\boldsymbol{q}_s$
  2. Determine the next neighbor $\boldsymbol{q}_{nn}$ in T
  3. Add points on the connection between $\boldsymbol{q}_s$ and $\boldsymbol{q}_{nn}$ to T
     - With step size d
     - Check every part of the path for collision with C_{obs}
     - Stop when a collision has been detected
  4. Go to 1.



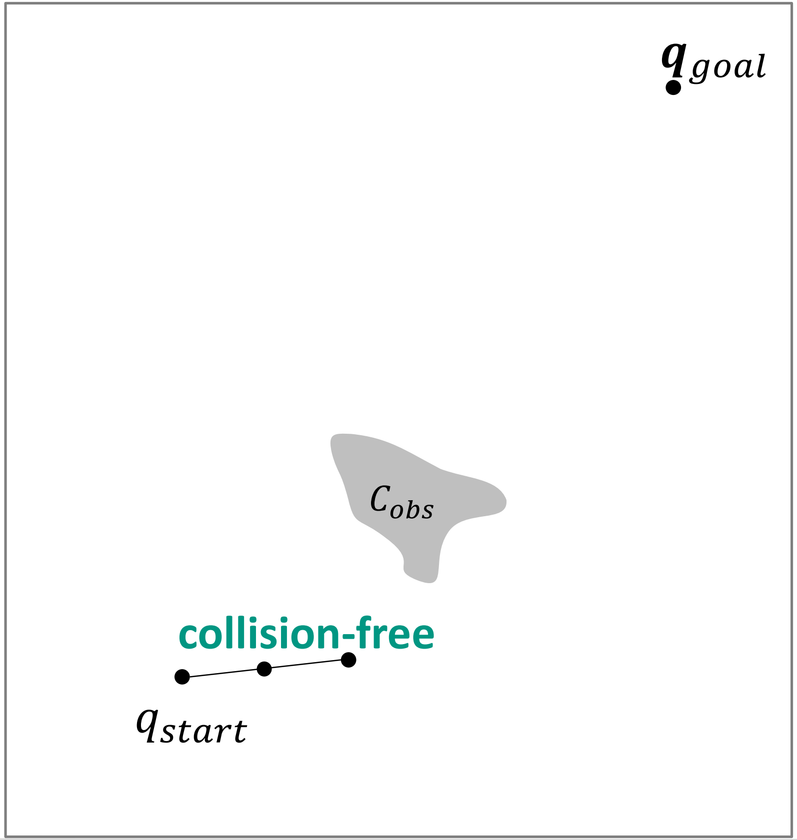
RRT: Principle (4)

- Iteration
  1. Sample a point $\boldsymbol{q}_s$
  2. Determine the next neighbor $\boldsymbol{q}_{nn}$ in T
  3. Add points on the connection between $\boldsymbol{q}_s$ and $\boldsymbol{q}_{nn}$ to T
     - With step size d
     - Check every part of the path for collision with C_{obs}
     - Stop when a collision has been detected
  4. Go to 1.



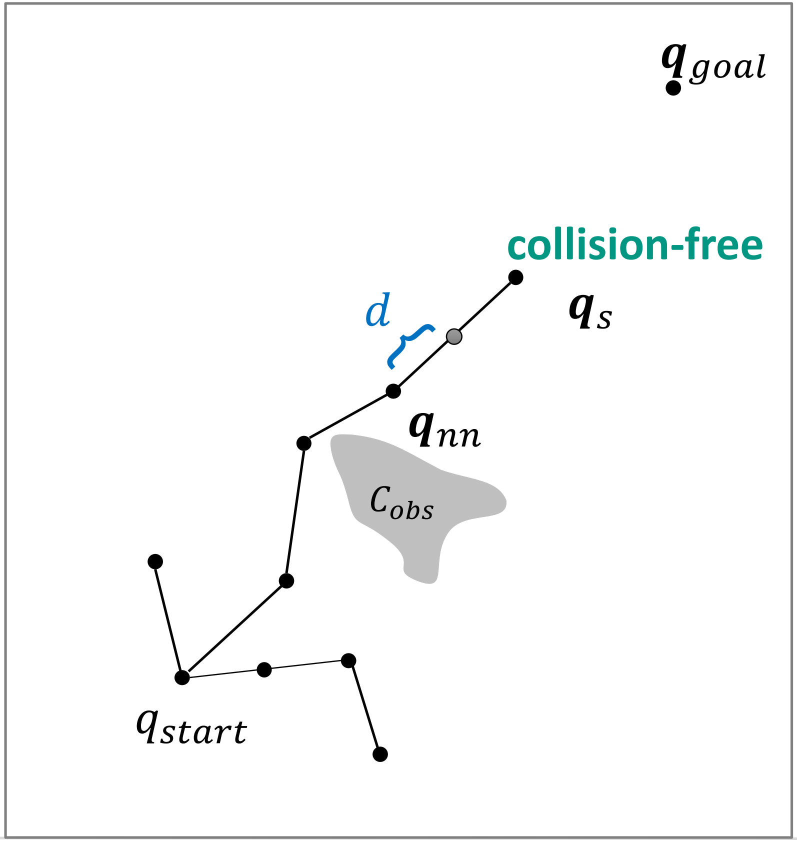
RRT: Principle (5)

- Iteration
  1. Sample a point $\boldsymbol{q}_s$
  2. Determine the next neighbor $\boldsymbol{q}_{nn}$ in T
  3. Add points on the connection between $\boldsymbol{q}_s$ and $\boldsymbol{q}_{nn}$ to T
     - With step size d
     - Check every part of the path for collision with C_{obs}
     - Stop when a collision has been detected
  4. Go to 1.



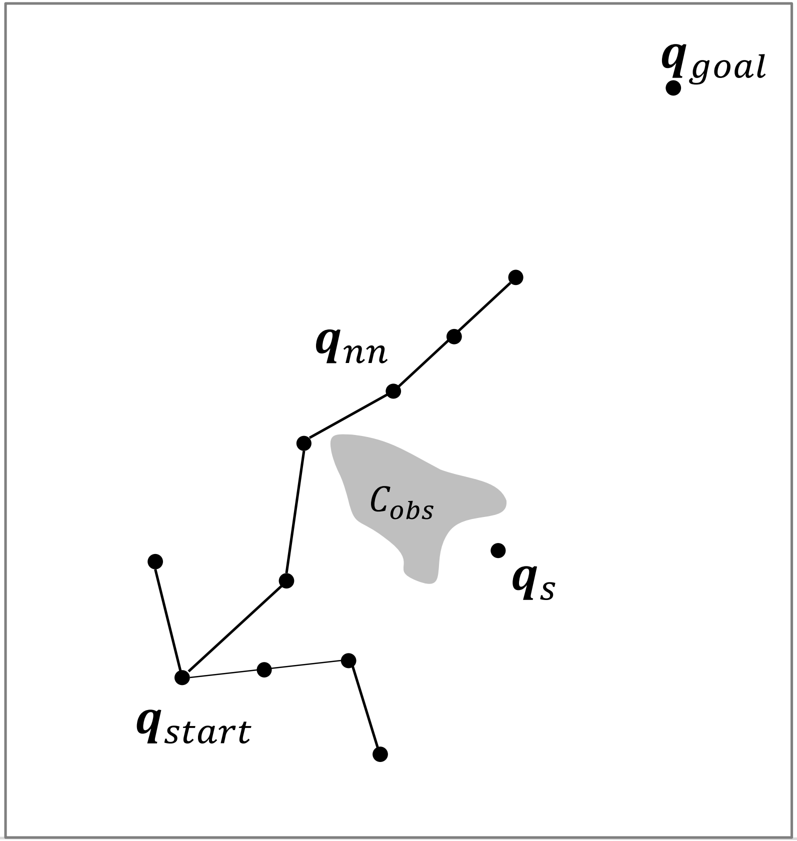
RRT: Principle (6)

- Iteration
  1. Sample a point $\boldsymbol{q}_s$
  2. Determine the next neighbor $\boldsymbol{q}_{nn}$ in T
  3. Add points on the connection between $\boldsymbol{q}_s$ and $\boldsymbol{q}_{nn}$ to T
     - With step size d
     - Check every part of the path for collision with C_{obs}
     - Stop when a collision has been detected
  4. Go to 1.



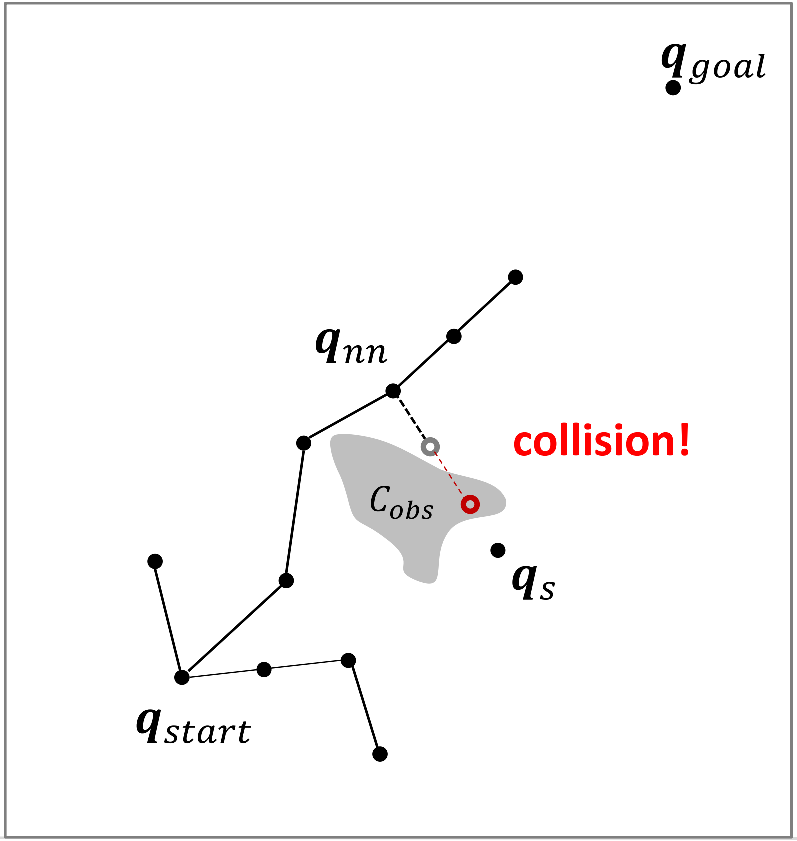
RRT: Principle (7)

- Iteration
  1. Sample a point $\boldsymbol{q}_s$
  2. Determine the next neighbor $\boldsymbol{q}_{nn}$ in T
  3. Add points on the connection between $\boldsymbol{q}_s$ and $\boldsymbol{q}_{nn}$ to T
     - With step size d
     - Check every part of the path for collision with C_{obs}
     - Stop when a collision has been detected
  4. Go to 1.



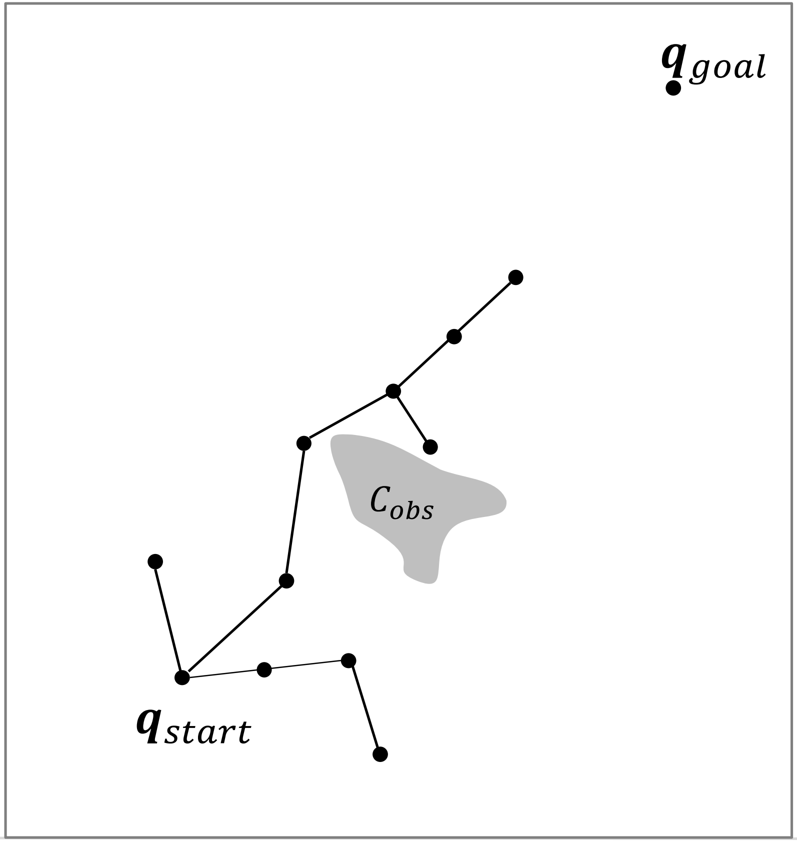
RRT: Principle (8)

- Iteration
  1. Sample a point $\boldsymbol{q}_s$
  2. Determine the next neighbor $\boldsymbol{q}_{nn}$ in T
  3. Add points on the connection between $\boldsymbol{q}_s$ and $\boldsymbol{q}_{nn}$ to T
     - With step size d
     - Check every part of the path for collision with C_{obs}
     - Stop when a collision has been detected
  4. Go to 1.



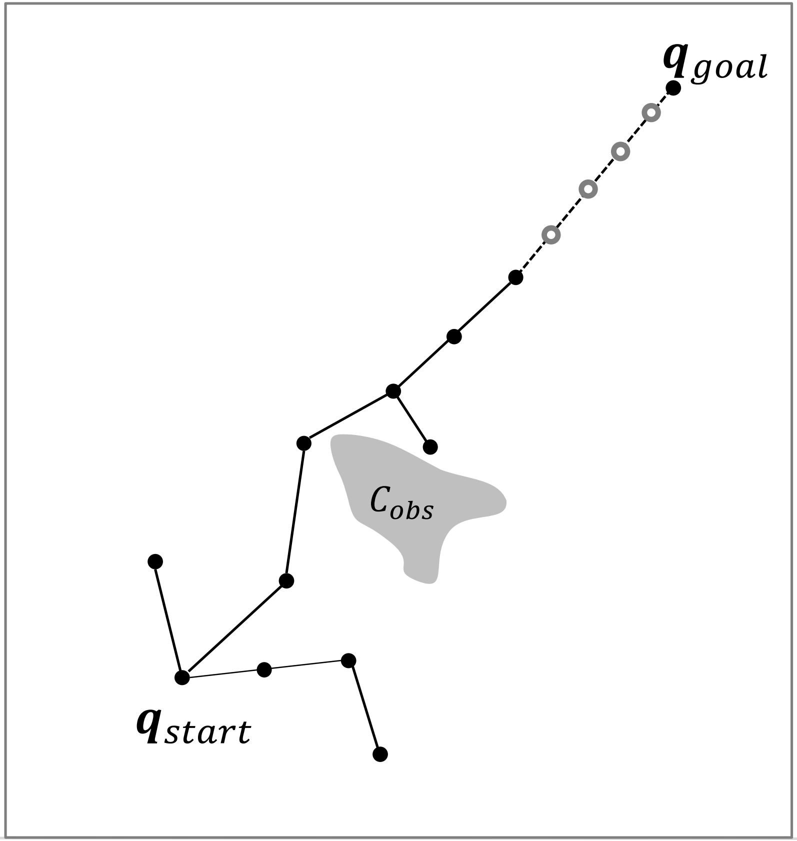
RRT: Principle (9)

- Iteration
  1. Sample a point $\boldsymbol{q}_s$
  2. Determine the next neighbor $\boldsymbol{q}_{nn}$ in T
  3. Add points on the connection between $\boldsymbol{q}_s$ and $\boldsymbol{q}_{nn}$ to T
     - With step size d
     - Check every part of the path for collision with C_{obs}
     - Stop when a collision has been detected
  4. Go to 1.



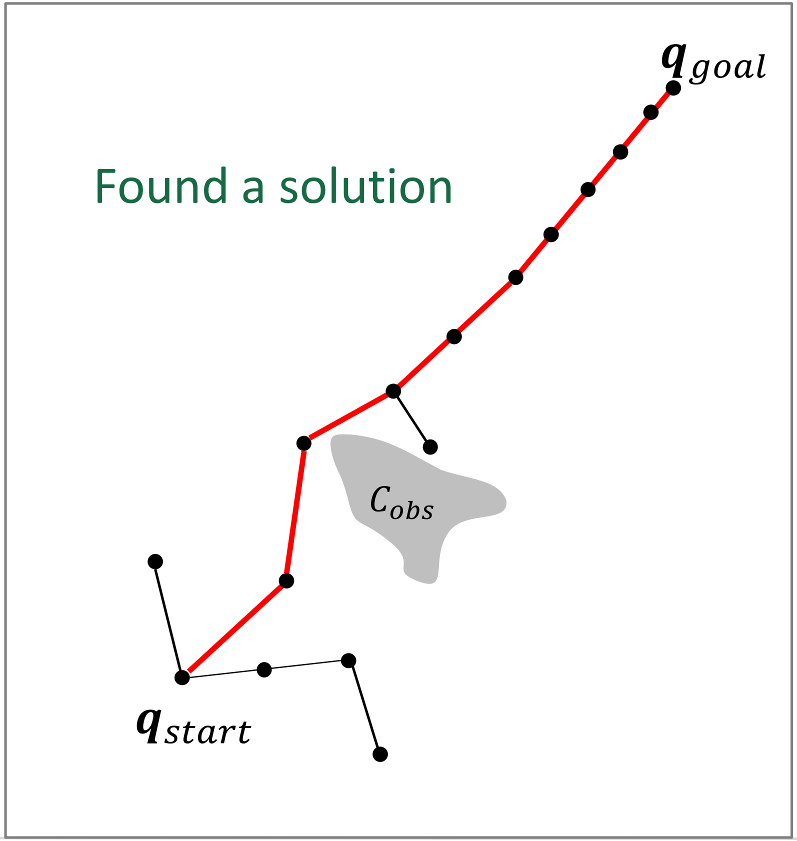
RRT: Principle (10)

- Iteration
  1. Sample a point $\boldsymbol{q}_s$
  2. Determine the next neighbor $\boldsymbol{q}_{nn}$ in T
  3. Add points on the connection between $\boldsymbol{q}_s$ and $\boldsymbol{q}_{nn}$ to T
     - With step size d
     - Check every part of the path for collision with C_{obs}
     - Stop when a collision has been detected
  4. Go to 1.
- Check in every k^{th} step whether $\boldsymbol{q}_{goal}$ can be connected to T



RTRT: Principle (11)

- Iteration
  1. Sample a point $\boldsymbol{q}_s$
  2. Determine the next neighbor $\boldsymbol{q}_{nn}$ in T
  3. Add points on the connection between $\boldsymbol{q}_s$ and $\boldsymbol{q}_{nn}$ to T
     - With step size d
     - Check every part of the path for collision with C_{obs}
     - Stop when a collision has been detected
  4. Go to 1.
- Check in every k^{th} step whether $\boldsymbol{q}_{goal}$ can be connected to T



RRT*

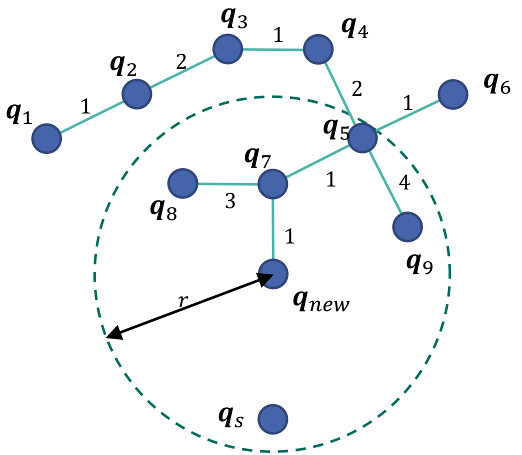
- **Problem:** RRTs yield trajectories that are usually not optimal
- **RRT*** optimizes the search space iteratively during the search

- Optimization of the search tree is divided into two steps:
  - Calculate costs of each new node (e.g., length of the path from the start node)
  - Rewiring of the search tree by adding new nodes
- **Disadvantage:**
  - Longer runtime (up to a factor of 30 in comparison to uni-directional RRT)
  - Uni-directional approach

S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, 30(7):846–894, Jan. 2011.

Exercise 3: RRT*

1. Explain how the node q_{new} was determined.
2. Calculate the path costs for the nodes $q_1, \dots, q_9, q_{new}$
3. Describe the RRT* function $Near(T, q_{new}, r)$
4. Which nodes are taken into account for the Rewire step?
5. Draw the connections after the Rewire step.



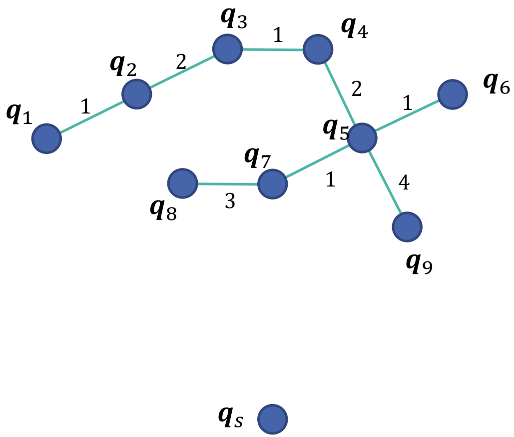
The tree T shows an intermediate step of RRT*

- Nodes $q_1, \dots, q_9$
- Connection costs indicated on the edges
- q_{new} added in the current iteration step

RRT*: Algorithm

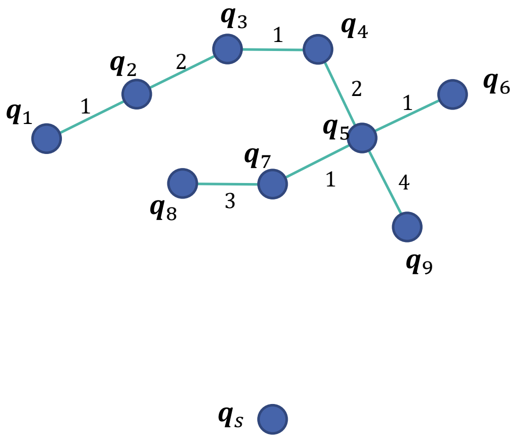
1. $\boldsymbol{q}_s = SampleRandom(C)$ // Sample random configuration
2. $\boldsymbol{q}_{nn} = NearestNeighbor(\boldsymbol{q}_s, T)$ // Determine the nearest neighbor
3. $\boldsymbol{q}_{new} = Steer(\boldsymbol{q}_{nn}, \boldsymbol{q}_s, d)$ // Go a step in the direction of $\boldsymbol{q}_s$
4. $if\ !CollisionFreePath(\boldsymbol{q}_{nn}, \boldsymbol{q}_{new}): goto\ 1$ // Is the path collision-free?
5. $Q_{near} = Near(T, \boldsymbol{q}_{new}, r)$ // All points with a distance to $\boldsymbol{q}_{new}$ of at most r
6. $\boldsymbol{q}_{min} = MinCostPath(Q_{near}, \boldsymbol{q}_{new})$ // $Cost(\boldsymbol{q}_{min}) + Cost(\boldsymbol{q}_{min}, \boldsymbol{q}_{new})$ minimal
7. $AddPath(T, \boldsymbol{q}_{min}, \boldsymbol{q}_{new})$ // Add path from $\boldsymbol{q}_{min}$ to $\boldsymbol{q}_{new}$
8. $Rewire(T, \boldsymbol{q}_{new}, Q_{near})$ // Check edges to nodes in Q_{near}
9. $if\ !Timeout: goto\ 1$ // Next iteration

Exercise 3.1: How was q_{new} determined? (1)



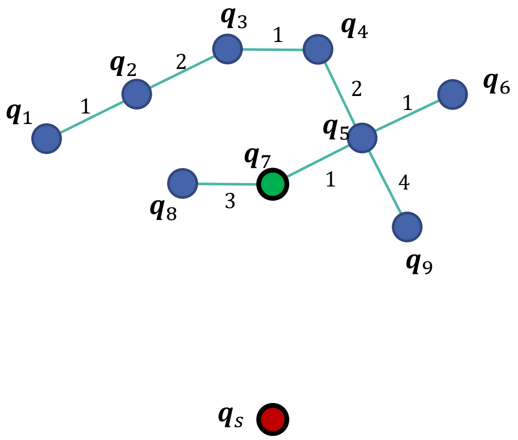
Exercise 3.1: How was q_{new} determined? (2)

- $\boldsymbol{q}_{new} = Steer(\boldsymbol{q}_{nn}, \boldsymbol{q}_s, d)$
  - $\boldsymbol{q}_s$: Current sample
  - $\boldsymbol{q}_{nn}$: Nearest neighbor to $\boldsymbol{q}_s$
  - d: Step size



Exercise 3.1: How was q_{new} determined? (3)

- $\boldsymbol{q}_{new} = Steer(\boldsymbol{q}_{nn}, \boldsymbol{q}_s, d)$
  - $\boldsymbol{q}_s$: Current sample
  - $\boldsymbol{q}_{nn}$: Nearest neighbor to $\boldsymbol{q}_s$
  - d: Step size

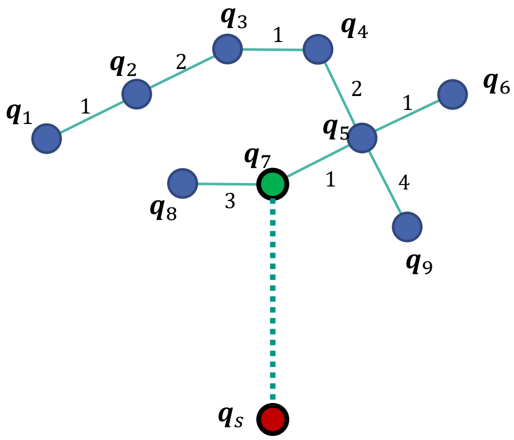


- (red) $\boldsymbol{q}_s$
- (green) $\boldsymbol{q}_{nn} = \boldsymbol{q}_7$

Exercise 3.1: How was q_{new} determined? (4)

- $\boldsymbol{q}_{new} = Steer(\boldsymbol{q}_{nn}, \boldsymbol{q}_s, d)$
  - $\boldsymbol{q}_s$: Current sample
  - $\boldsymbol{q}_{nn}$: Nearest neighbor to $\boldsymbol{q}_s$
  - d: Step size

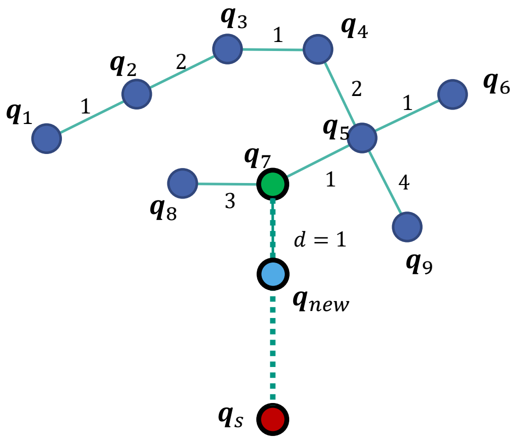
- $\boldsymbol{q}_s$
- $\boldsymbol{q}_{nn} = \boldsymbol{q}_7$
- Connection from $\boldsymbol{q}_{nn}$ to $\boldsymbol{q}_s$



Exercise 3.1: How was q_{new} determined? (5)

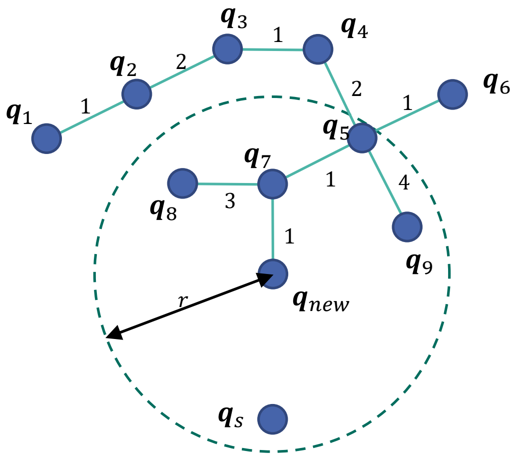
- $\boldsymbol{q}_{new} = Steer(\boldsymbol{q}_{nn}, \boldsymbol{q}_s, d)$
  - $\boldsymbol{q}_s$: Current sample
  - $\boldsymbol{q}_{nn}$: Nearest neighbor to $\boldsymbol{q}_s$
  - d: Step size

- ● $\boldsymbol{q}_s$
- ● $\boldsymbol{q}_{nn} = \boldsymbol{q}_7$
- ····· Connection from $\boldsymbol{q}_{nn}$ to $\boldsymbol{q}_s$
- — Connection with step size $d = 1$
- ● $\boldsymbol{q}_{new}$



Exercise 3: RRT*

1. Explain how the node q_{new} was determined.
2. Calculate the path costs for the nodes $q_1, \dots, q_9, q_{new}$
3. Describe the RRT* function $Near(T, q_{new}, r)$
4. Which nodes are taken into account for the Rewire step?
5. Draw the connections after the Rewire step.

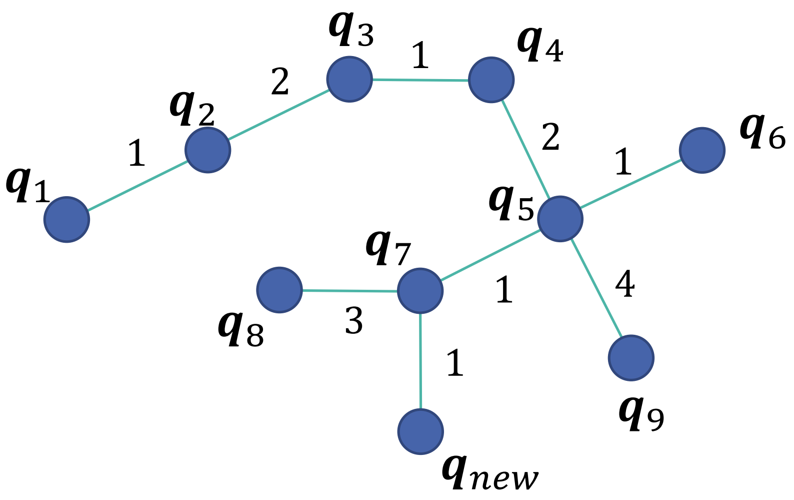


The tree T shows an intermediate step of RRT*

- Nodes $q_1, \dots, q_9$
- Connection costs indicated on the edges
- q_{new} added in the current iteration step

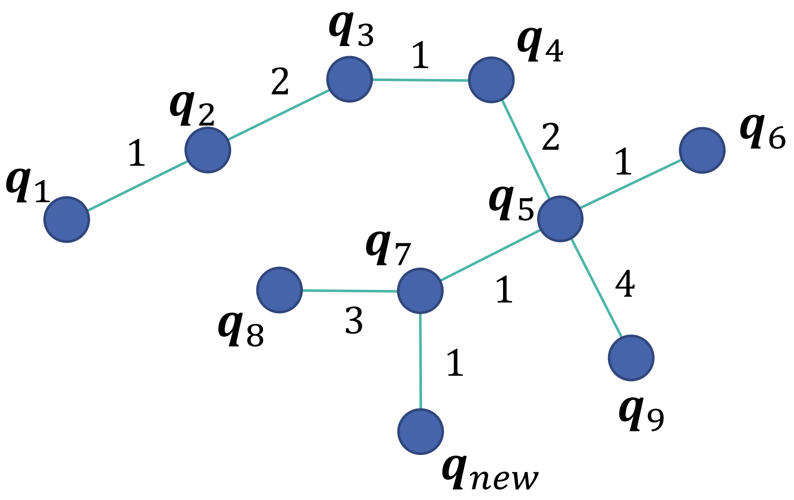
Exercise 3.2: Path costs for $q_1, \ldots, q_9, q_{new}$ (1)

Node	Path costs
$q_1 = q_{start}$	
q_2	
q_3	
q_4	
q_5	
q_6	
q_7	
q_8	
q_9	
q_{new}	



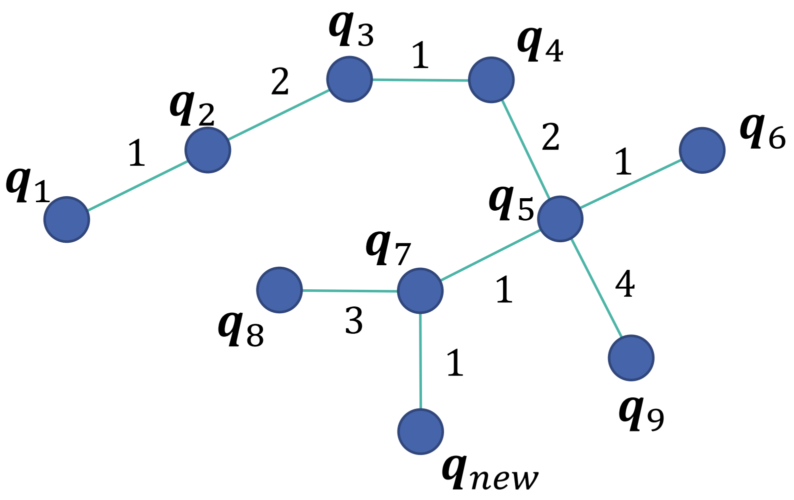
Exercise 3.2: Path costs for $q_1, \dots, q_9, q_{new}$ (2)

Node	Path costs
$\boldsymbol{q}_1$	0
$\boldsymbol{q}_2$	$c(\boldsymbol{q}_1) + 1 = 0 + 1 = 1$
$\boldsymbol{q}_3$	$c(\boldsymbol{q}_2) + 2 = 1 + 2 = 3$
$\boldsymbol{q}_4$	$c(\boldsymbol{q}_3) + 1 = 3 + 1 = 4$
$\boldsymbol{q}_5$	$c(\boldsymbol{q}_4) + 2 = 4 + 2 = 6$
$\boldsymbol{q}_6$	
$\boldsymbol{q}_7$	
$\boldsymbol{q}_8$	
$\boldsymbol{q}_9$	
$\boldsymbol{q}_{new}$	



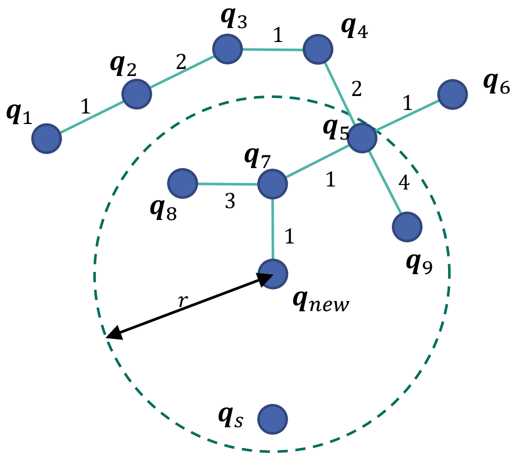
Exercise 3.2: Path costs for $q_1, \dots, q_9, q_{new}$ (3)

Node	Path costs
$\boldsymbol{q}_1$	0
$\boldsymbol{q}_2$	1
$\boldsymbol{q}_3$	3
$\boldsymbol{q}_4$	4
$\boldsymbol{q}_5$	6
$\boldsymbol{q}_6$	$c(\boldsymbol{q}_5) + 1 = 6 + 1 = 7$
$\boldsymbol{q}_7$	$c(\boldsymbol{q}_5) + 1 = 6 + 1 = 7$
$\boldsymbol{q}_8$	$c(\boldsymbol{q}_7) + 3 = 7 + 3 = 10$
$\boldsymbol{q}_9$	$c(\boldsymbol{q}_5) + 4 = 6 + 4 = 10$
$\boldsymbol{q}_{new}$	$c(\boldsymbol{q}_7) + 1 = 7 + 1 = 8$



Exercise 3: RRT*

1. Explain how the node q_{new} was determined.
2. Calculate the path costs for the nodes $q_1, \dots, q_9, q_{new}$
3. Describe the RRT* function $Near(T, q_{new}, r)$
4. Which nodes are taken into account for the Rewire step?
5. Draw the connections after the Rewire step.

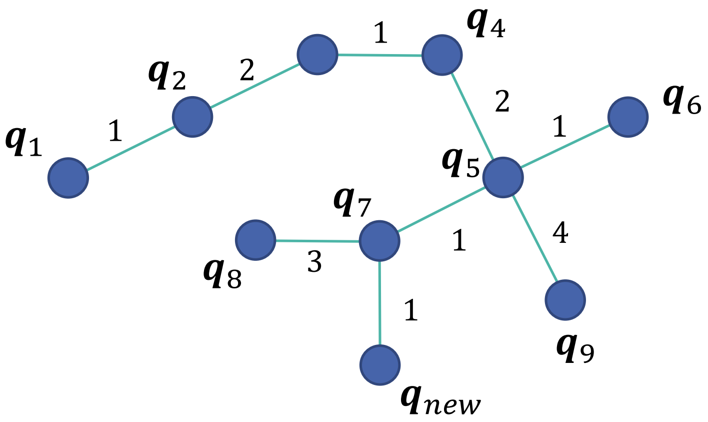


The tree T shows an intermediate step of RRT*

- Nodes $q_1, \dots, q_9$
- Connection costs indicated on the edges
- q_{new} added in the current iteration step

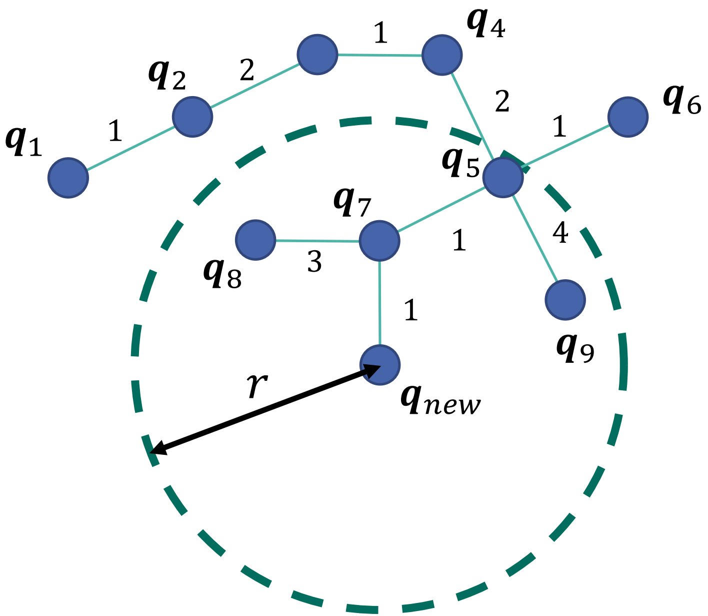
Exercise 3.3: RRT* Function $Near(T, q_{new}, r)$ (1)

- $Q_{near} = Near(T, q_{new}, r)$



Exercise 3.3: RRT* Function $Near(T, q_{new}, r)$ (2)

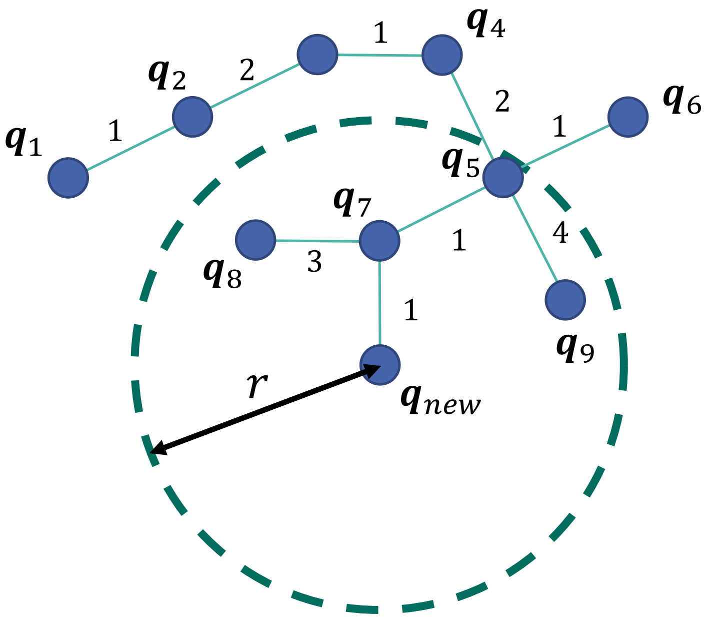
- $Q_{near} = Near(T, q_{new}, r)$
  - T: Tree
  - q_{new}: New node in the tree
  - r: Max. distance to determine neighboring nodes
  - Q_{near}: Set of neighboring nodes with distance $< r$



Exercise 3.3: RRT* Function $Near(T, q_{new}, r)$ (3)

- $Q_{near} = Near(T, q_{new}, r)$
  - T: Tree
  - q_{new}: New node in the tree
  - r: Max. distance to determine neighboring nodes
  - Q_{near}: Set of neighboring nodes with distance $< r$

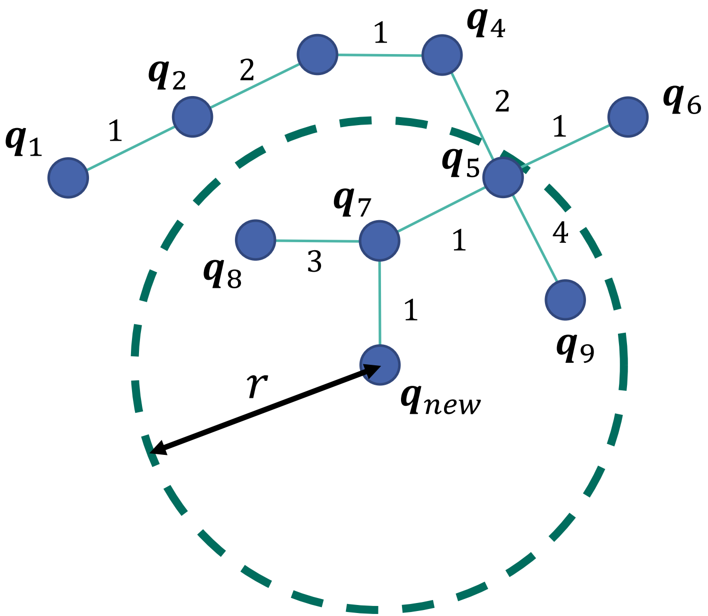
- $Near(T, q_{new}, r)$ determines all nodes from T whose distance to q_{new} is at most r.



Exercise 3.4: Nodes for the Rewire step (1)

- $Q_{near} = Near(T, q_{new}, r)$
  - T: Tree
  - q_{new}: New node in the tree
  - r: Max. distance to determine neighboring nodes
  - Q_{near}: Set of neighboring nodes with distance $< r$

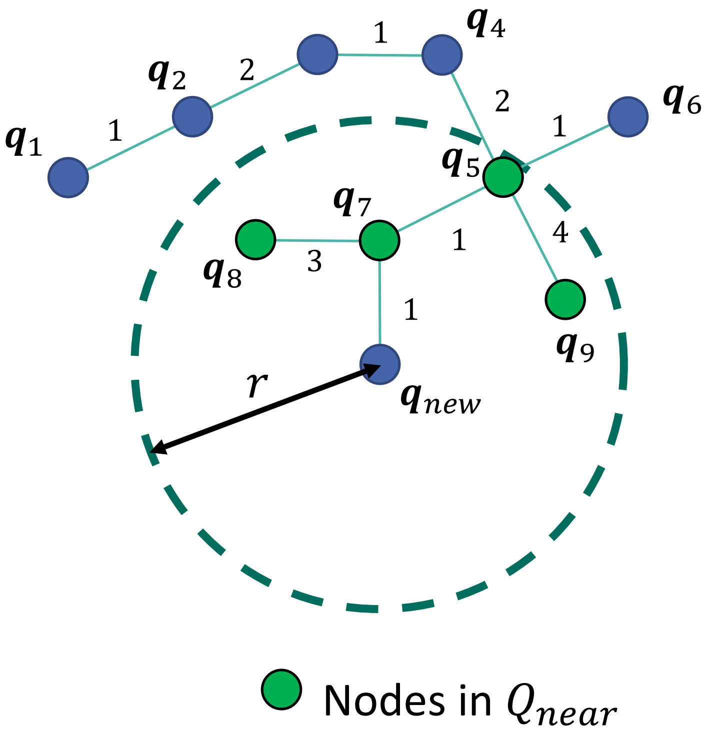
- $Q_{near} = \{$



Exercise 3.4: Nodes for the Rewire step (2)

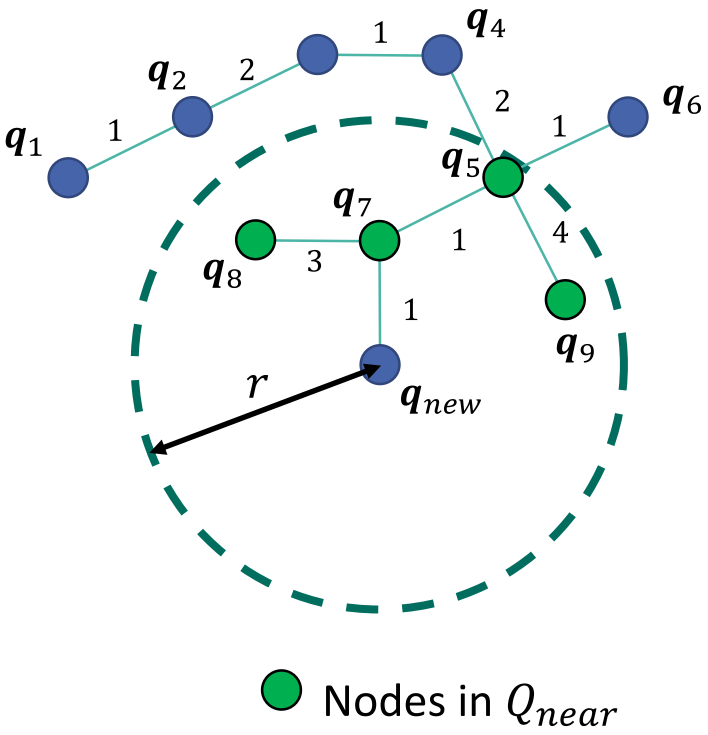
- $Q_{near} = Near(T, q_{new}, r)$
  - T: Tree
  - q_{new}: New node in the tree
  - r: Max. distance to determine neighboring nodes
  - Q_{near}: Set of neighboring nodes with distance $< r$

- $Q_{near} = \{\boldsymbol{q}_5,\ \boldsymbol{q}_7,\ \boldsymbol{q}_8,\ \boldsymbol{q}_9\}$



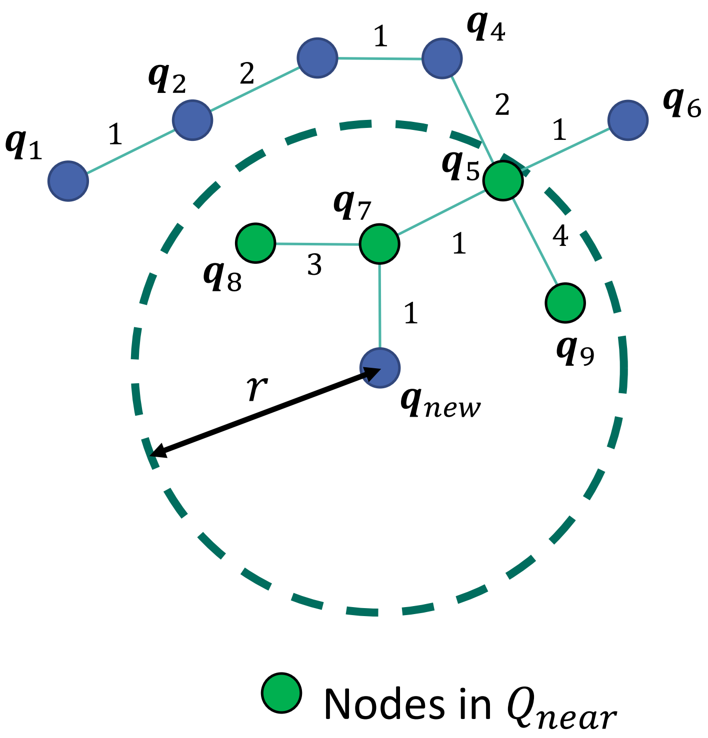
Exercise 3.5: Rewire step (1)

- $Q_{near} = \{\boldsymbol{q}_5, \ \boldsymbol{q}_7, \ \boldsymbol{q}_8, \ \boldsymbol{q}_9\}$



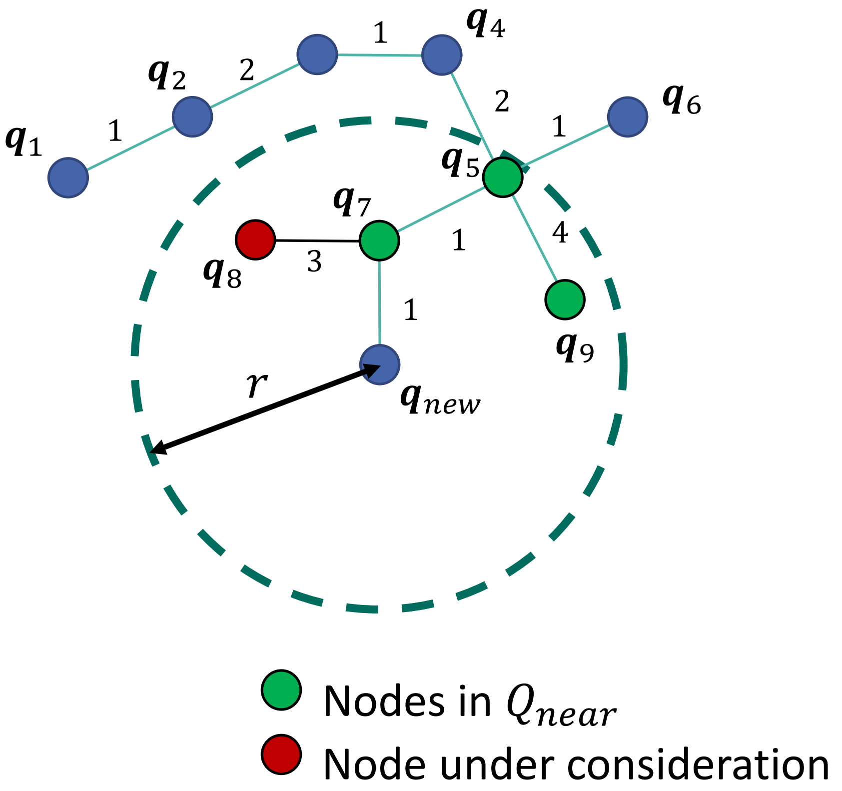
Exercise 3.5: Rewire step (2)

- $Q_{near} = \{\boldsymbol{q}_5, \boldsymbol{q}_7, \boldsymbol{q}_8, \boldsymbol{q}_9\}$
- $\boldsymbol{q}_7$ is already connected to $\boldsymbol{q}_{new}$
- $\boldsymbol{q}_5$ is part of the path with the minimum cost to $\boldsymbol{q}_1$



Exercise 3.5: Rewire step (3)

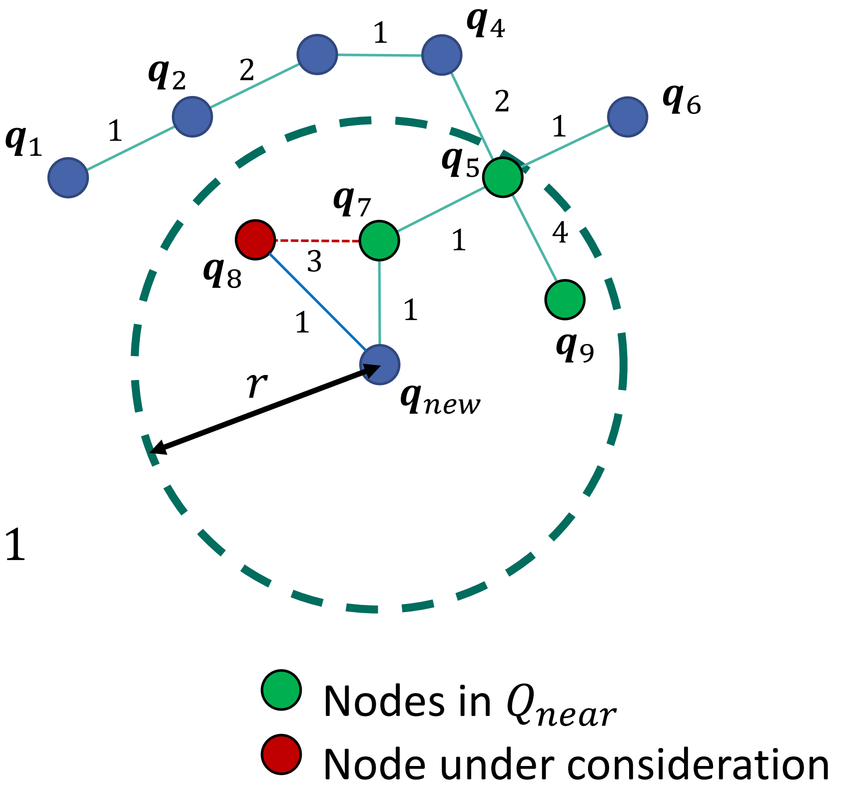
- $Q_{near} = \{\boldsymbol{q}_5, \ \boldsymbol{q}_7, \ \boldsymbol{q}_8, \ \boldsymbol{q}_9\}$
- Costs
  - $Cost(\boldsymbol{q}_{new}) = 8$
  - $Cost(\boldsymbol{q}_5) = 6$
  - $Cost(\boldsymbol{q}_8) = 10$
  - $Cost(\boldsymbol{q}_9) = 10$
  - $Cost(\boldsymbol{q}_{new}, \ \boldsymbol{q}_8) = 1$
- Rewire $\boldsymbol{q}_8$:



Exercise 3.5: Rewire step (4)

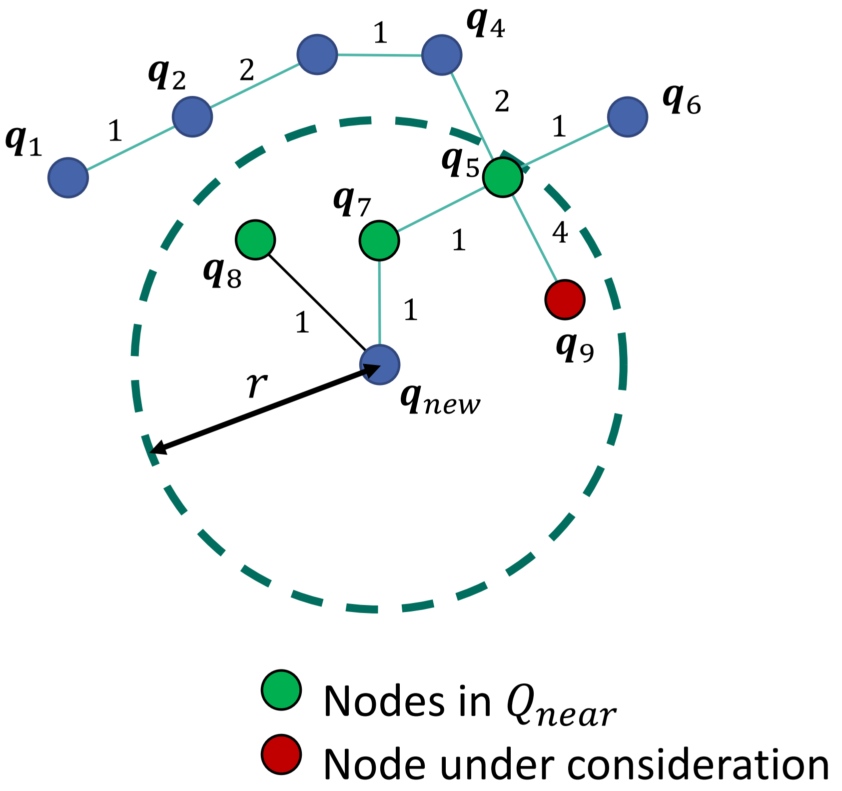
- $Q_{near} = \{\boldsymbol{q}_5, \boldsymbol{q}_7, \boldsymbol{q}_8, \boldsymbol{q}_9\}$
- Costs
  - $Cost(\boldsymbol{q}_{new}) = 8$
  - $Cost(\boldsymbol{q}_5) = 6$
  - $Cost(\boldsymbol{q}_8) = 10$
  - $Cost(\boldsymbol{q}_9) = 10$
  - $Cost(\boldsymbol{q}_{new}, \boldsymbol{q}_8) = 1$
- Rewire $\boldsymbol{q}_8$:
  - $Cost(\boldsymbol{q}_{new}) + Cost(\boldsymbol{q}_{new}, \boldsymbol{q}_8) = 8 + 1$
  - $9 < Cost(\boldsymbol{q}_8) = 10$

    ➔ Rewiring



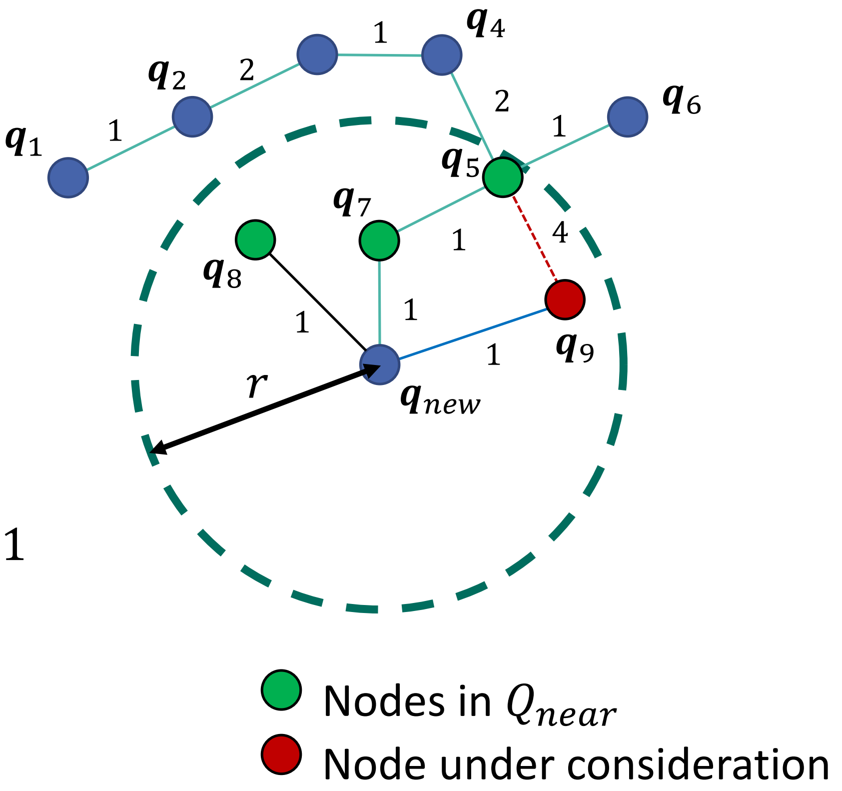
Exercise 3.5: Rewire step (5)

- $Q_{near} = \{\boldsymbol{q}_5, \boldsymbol{q}_7, \boldsymbol{q}_8, \boldsymbol{q}_9\}$
- Costs
  - $Cost(\boldsymbol{q}_{new}) = 8$
  - $Cost(\boldsymbol{q}_5) = 6$
  - $Cost(\boldsymbol{q}_8) = 10$
  - $Cost(\boldsymbol{q}_9) = 10$
  - $Cost(\boldsymbol{q}_{new}, \boldsymbol{q}_9) = 1$
- Rewire $\boldsymbol{q}_9$:



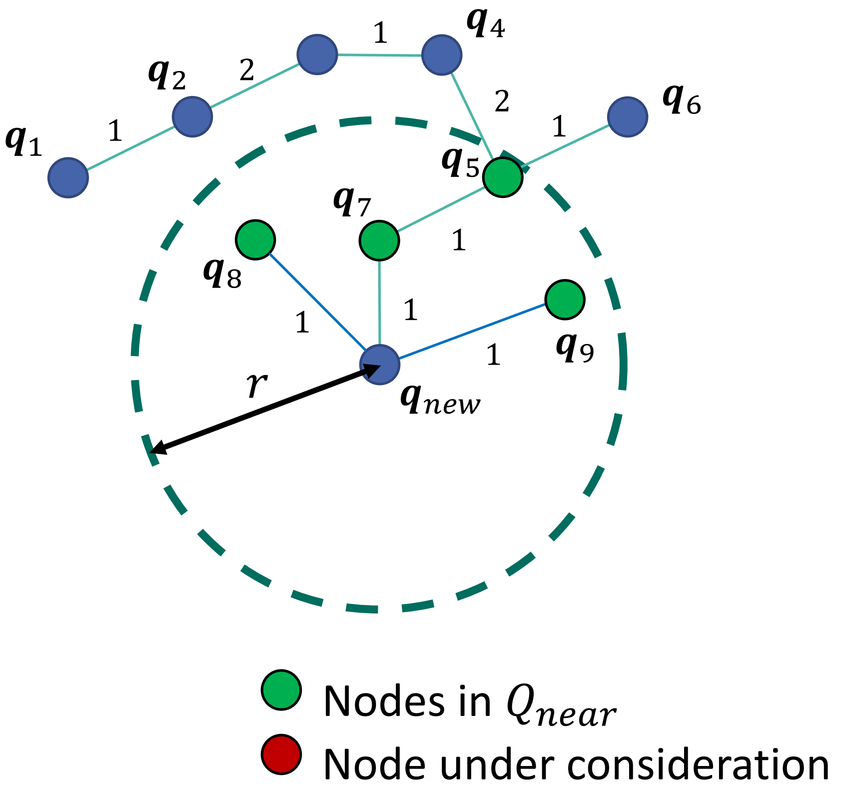
Exercise 3.5: Rewire step (6)

- $Q_{near} = \{\boldsymbol{q}_5, \boldsymbol{q}_7, \boldsymbol{q}_8, \boldsymbol{q}_9\}$
- Costs
  - $Cost(\boldsymbol{q}_{new}) = 8$
  - $Cost(\boldsymbol{q}_5) = 6$
  - $Cost(\boldsymbol{q}_8) = 10$
  - $Cost(\boldsymbol{q}_9) = 10$
  - $Cost(\boldsymbol{q}_{new}, \boldsymbol{q}_9) = 1$
- Rewire $\boldsymbol{q}_9$:
  - $Cost(\boldsymbol{q}_{new}) + Cost(\boldsymbol{q}_{new}, \boldsymbol{q}_9) = 8 + 1$
  - $9 < Cost(\boldsymbol{q}_9) = 10$
    ➔ Rewiring



Exercise 3.5: Rewire step (7)

- $Q_{near} = \{\boldsymbol{q}_5,\ \boldsymbol{q}_7,\ \boldsymbol{q}_8,\ \boldsymbol{q}_9\}$
- Costs
  - $Cost(\boldsymbol{q}_{new}) = 8$
  - $Cost(\boldsymbol{q}_5) = 6$
  - $Cost(\boldsymbol{q}_8) = 10$
  - $Cost(\boldsymbol{q}_9) = 10$
- Outcome:
  - Rewire: $\boldsymbol{q}_{new} \rightarrow \boldsymbol{q}_8$
  - Rewire: $\boldsymbol{q}_{new} \rightarrow \boldsymbol{q}_9$



Exercise 3: RRT*, Bonus

- How does the RRT* algorithm differ from the RRT algorithm?

  a) Unlike RRT, RRT* does not require any preprocessing

  b) The search tree is optimized iteratively during the search

  c) The search is conducted from both sides ($\boldsymbol{q}_{start}$ and $\boldsymbol{q}_{goal}$)

Exercise 3: RRT*, Bonus

- How does the RRT* algorithm differ from the RRT algorithm?

  a) Unlike RRT, RRT* does not require any preprocessing

  b) The search tree is optimized iteratively during the search
  Correct: This is the purpose of the “Rewire” step, see previous slides

  c) The search is conducted from both sides ($\boldsymbol{q}_{start}$ and $\boldsymbol{q}_{goal}$)

Exercise 3: RRT*, Bonus

- How does the RRT* algorithm differ from the RRT algorithm?

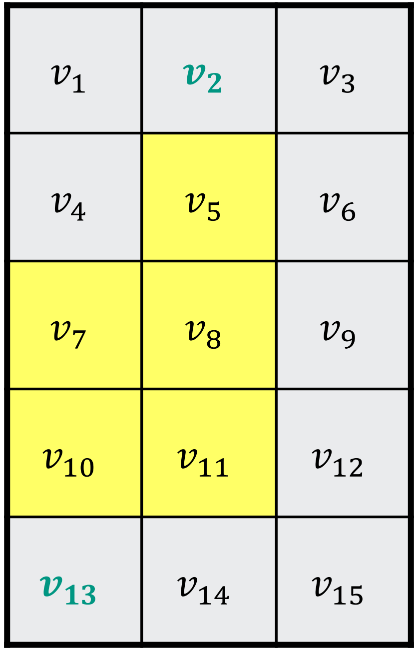
a) Unlike RRT, RRT* does not require any preprocessing
Wrong: Neither RRT nor RRT* require preprocessing.
However, preprocessing is required for probabilistic road maps.

b) The search tree is optimized iteratively during the search
Correct: This is the purpose of the "Rewire" step, see previous slides

c) The search is conducted from both sides ($\boldsymbol{q}_{start}$ and $\boldsymbol{q}_{goal}$)
Wrong: This applies to bidirectional RRTs like RRT Connect

Exercise 4: A*-Algorithm

1. Find the optimal path from v_2 to v_{13}
   - Only horizontal and vertical movements allowed
   - Costs:
     - Entering a grey cell: 1
     - Entering a yellow cell: 4
   - Heuristic h:
     Euclidean distance to v_{13}
     (e.g. from v_{11} to v_{13}: $h(v_{11}) = \sqrt{2}$)
2. Why is the Euclidean distance a suitable heuristic?
3. When does the A* algorithm find a valid solution?



A*- Algorithm (1)

- Iterative approach
- Two node sets:
  - **Open Set** O: nodes not visited yet
  - **Closed Set** C: nodes already visited
- **Update**: for a visited node v_n:
  - **Predecessor node** $pred(v_n)$
  - **Accumulated cost** to reach v_n: $g(v_n)$
  - **Total cost** $f(v_n) = g(v_n) + h(v_n)$, with $h(v_n)$ being a heuristic estimating the cost to v_{goal}
- **Initialize**
  - $O = \{v_s\}$
  - $C = \{\}$
  - $g(v_i) = \infty \quad 1 \leq i \leq K$
  - $g(v_s) = 0$

A*- Algorithm (2)

- **Algorithm**
  while $O \neq \emptyset$
  - Determine next node to expand
    - find $v_i \in O$ with minimal $f(v_i) = g(v_i) + h(v_i)$
  - if $v_i = v_{goal}$
    found solution: traverse predecessor of v_i until v_{start} is reached
  - $O.remove(v_i)$
  - $C.add(v_i)$
  - **Update** all successors v_j of v_i

    if $v_j \in C$, skip v_j

    if $v_j \notin O$, $O.add(v_j)$

    if $g(v_i) + cost(v_i, v_j) < g(v_j)$

    $$g(v_j) = g(v_i) + cost(v_i, v_j)$$
    $$h(v_j) = heuristic(v_j, v_{goal})$$
    $$pred(v_j) = v_i$$

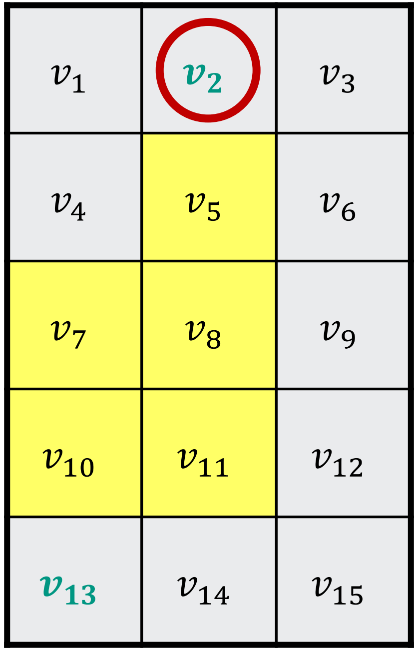
Exercise 4.1: A*- Algorithm, Initialization

- Initialization:
  - $O = \{v_2\}$
    - $f(v_2) = 0 + h(v_2) = \sqrt{4^2 + 1^2} \approx 4.12$
  - $C = \{\ \}$

v_1	v_2	v_3
v_4	v_5	v_6
v_7	v_8	v_9
v_{10}	v_{11}	v_{12}
v_{13}	v_{14}	v_{15}

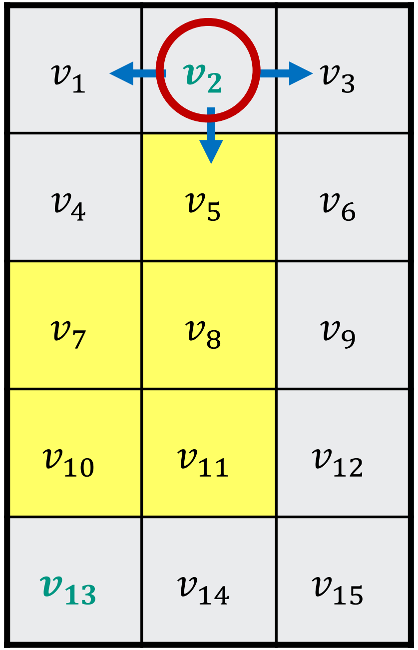
Exercise 4.1: A*- Algorithm, Step 1 (1)

- State:
  - $O = \{v_2\}$
    - $f(v_2) = 0 + h(v_2) = \sqrt{4^2 + 1^2} \approx 4.12$
  - $C = \{\,\}$
- Update:
  - Expand $\boldsymbol{v_2}$



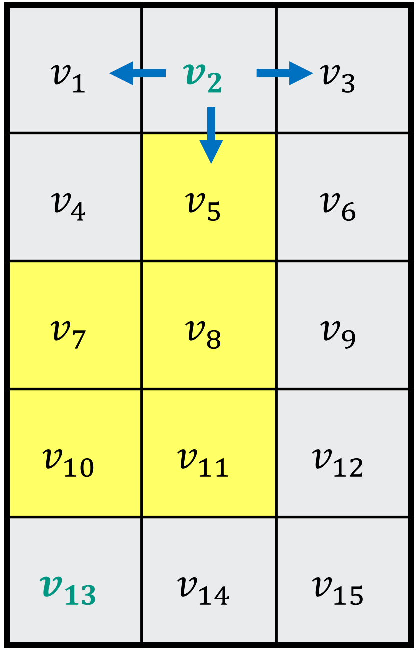
Exercise 4.1: A*- Algorithm, Step 1 (2)

- State:
  - $O = \{v_2\}$
    - $f(v_2) = 0 + h(v_2) = \sqrt{4^2 + 1^2} \approx 4.12$
  - $C = \{\,\}$
- Update:
  - Expand $\boldsymbol{v_2}$
  - $O = \{\boldsymbol{v_1}, \boldsymbol{v_3}, \boldsymbol{v_5}\}$
    - $f(v_1) = 1 + h(v_1) = 1 + 4 = 5$
    - $f(v_3) = 1 + h(v_3) = 1 + \sqrt{4^2 + 2^2} \approx 5.47$
    - $f(v_5) = 4 + h(v_5) = 4 + \sqrt{3^2 + 1^2} \approx 7.16$
  - $C = \{\boldsymbol{v_2}\}$



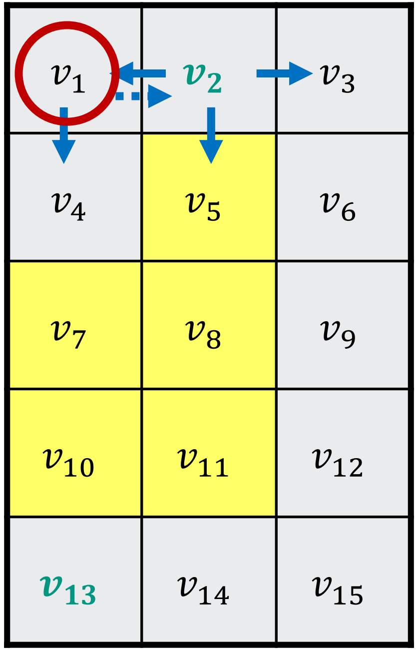
Exercise 4.1: A*- Algorithm, Step 2 (1)

- State:
  - $O = \{v_1, v_3, v_5\}$
    - $f(v_1) = 5$
    - $f(v_3) \approx 5.47$
    - $f(v_5) \approx 7.16$
  - $C = \{v_2\}$
- Update:



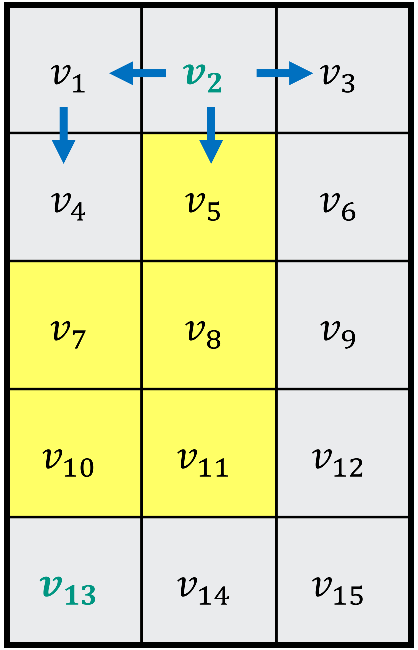
Exercise 4.1: A*- Algorithm, Step 2 (2)

- State:
  - $O = \{v_1, v_3, v_5\}$
    - $f(v_1) = 5$
    - $f(v_3) \approx 5.47$
    - $f(v_5) \approx 7.16$
  - $C = \{v_2\}$
- Update:
  - Expand $\boldsymbol{v_1}$
  - $O = \{v_3, v_5, \boldsymbol{v_4}\}$
    - $f(v_4) = 2 + h(v_4) = 2 + 3 = 5$
  - $C = \{v_2, \boldsymbol{v_1}\}$



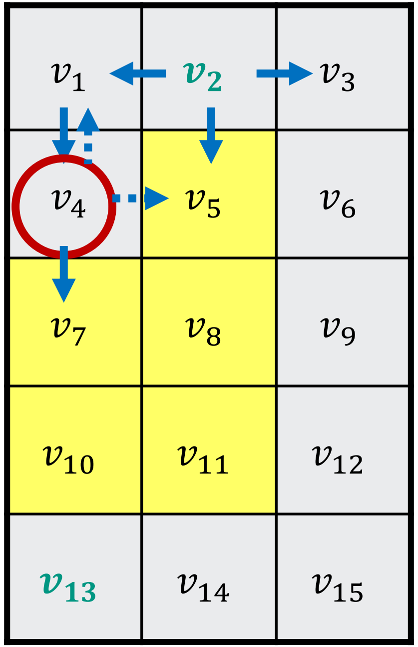
Exercise 4.1: A*- Algorithm, Step 3 (1)

- State:
  - $O = \{v_3, v_4, v_5\}$
    - $f(v_3) \approx 5.47$
    - $f(v_4) = 5$
    - $f(v_5) \approx 7.16$
  - $C = \{v_1, v_2\}$
- Update:



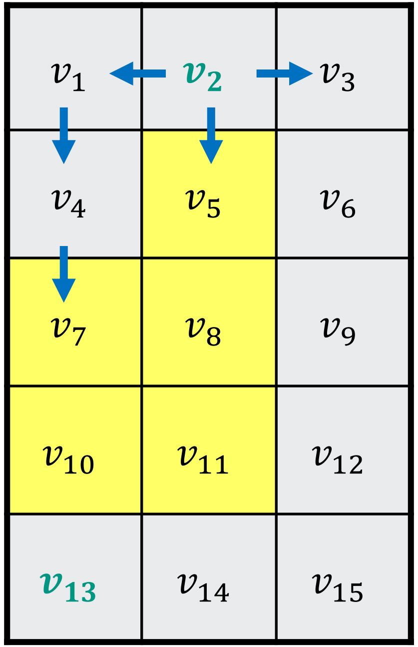
Exercise 4.1: A*- Algorithm, Step 3 (2)

- State:
  - $O = \{v_3, v_4, v_5\}$
    - $f(v_3) \approx 5.47$
    - $f(v_4) = 5$
    - $f(v_5) \approx 7.16$
  - $C = \{v_1, v_2\}$
- Update:
  - Expand $\boldsymbol{v_4}$
  - $O = \{v_3, \boldsymbol{v_5}, \boldsymbol{v_7}\}$
    - $g(v_4) + cost(v_4, v_5) = 2 + 4 = 6 \geq g(v_5) = 4$
      $\Rightarrow$ No update
    - $f(v_7) = 6 + h(v_7) = 6 + 2 = 8$
  - $C = \{v_1, v_2, \boldsymbol{v_4}\}$



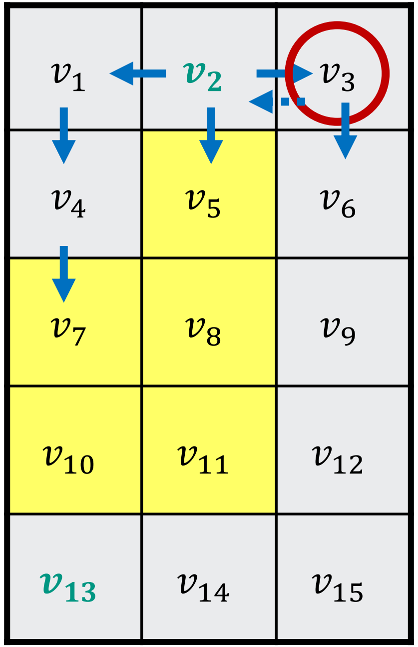
Exercise 4.1: A*- Algorithm, Step 4 [optional] (1)

- State:
  - $O = \{v_3, v_5, v_7\}$
    - $f(v_3) \approx 5.47$
    - $f(v_5) \approx 7.16$
    - $f(v_7) = 8$
  - $C = \{v_1, v_2, v_4\}$
- Update:



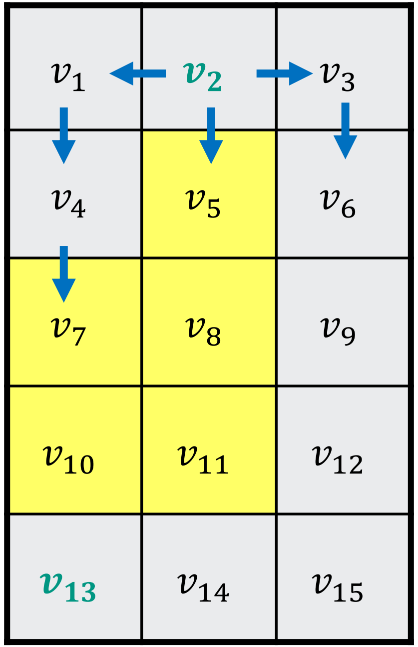
Exercise 4.1: A*- Algorithm, Step 4 [optional] (2)

- State:
  - $O = \{v_3, v_5, v_7\}$
    - $f(v_3) \approx 5.47$
    - $f(v_5) \approx 7.16$
    - $f(v_7) = 8$
  - $C = \{v_1, v_2, v_4\}$
- Update:
  - Expand $\boldsymbol{v_3}$
  - $O = \{v_5, \boldsymbol{v_6}, v_7\}$
    - $f(v_6) = 2 + h(v_6) = 2 + \sqrt{3^2 + 2^2} \approx 5.61$
  - $C = \{v_1, v_2, v_4, \boldsymbol{v_3}\}$



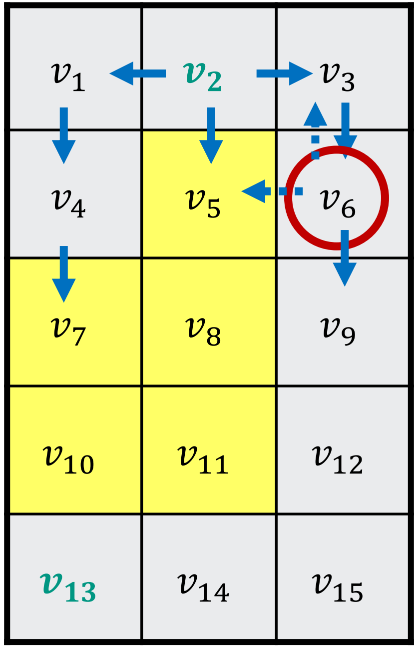
Exercise 4.1: A*- Algorithm, Step 5 [optional] (1)

- State:
  - $O = \{v_5, v_6, v_7\}$
    - $f(v_5) \approx 7.16$
    - $f(v_6) \approx 5.47$
    - $f(v_7) = 8$
  - $C = \{v_1, v_2, v_3, v_4\}$
- Update:



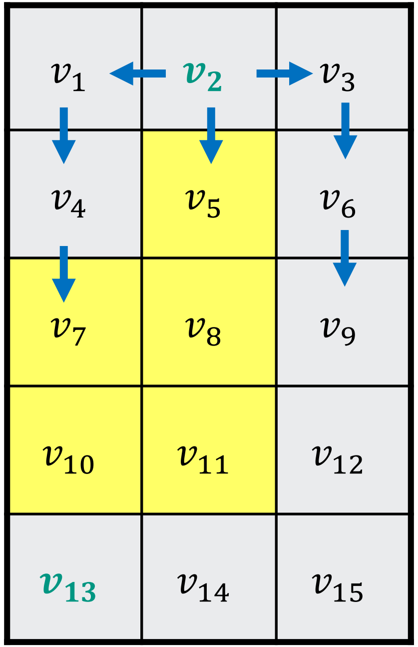
Exercise 4.1: A*- Algorithm, Step 5 [optional] (2)

- State:
  - $O = \{v_5, v_6, v_7\}$
    - $f(v_5) \approx 7.16$
    - $f(v_6) \approx 5.47$
    - $f(v_7) = 8$
  - $C = \{v_1, v_2, v_3, v_4\}$
- Update:
  - Expand $\boldsymbol{v_6}$
  - $O = \{\boldsymbol{v_5}, v_7, \boldsymbol{v_9}\}$
    - $g(v_6) + cost(v_6, v_5) = 2 + 4 = 6 \geq g(v_5) = 4$
      $\Rightarrow$ Kein Update
    - $f(v_9) = 3 + h(v_9) = 3 + \sqrt{2^2 + 2^2} \approx 5.83$
  - $C = \{v_1, v_2, v_3, v_4, \boldsymbol{v_6}\}$



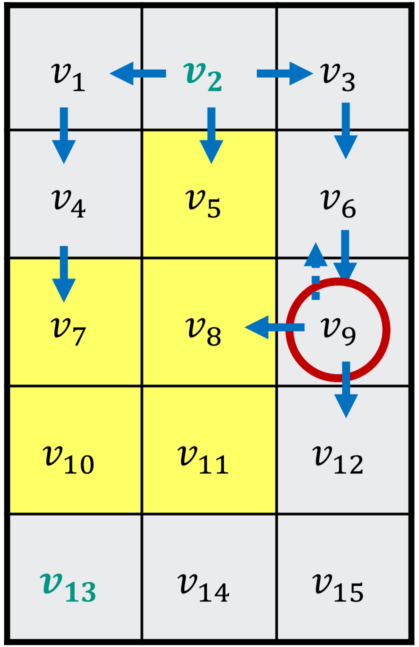
Exercise 4.1: A*- Algorithm, Step 6 [optional] (1)

- State:
  - $O = \{v_5, v_7, v_9\}$
    - $f(v_5) \approx 7.16$
    - $f(v_7) = 8$
    - $f(v_9) \approx 5.83$
  - $C = \{v_1, v_2, v_3, v_4, v_6\}$
- Update:



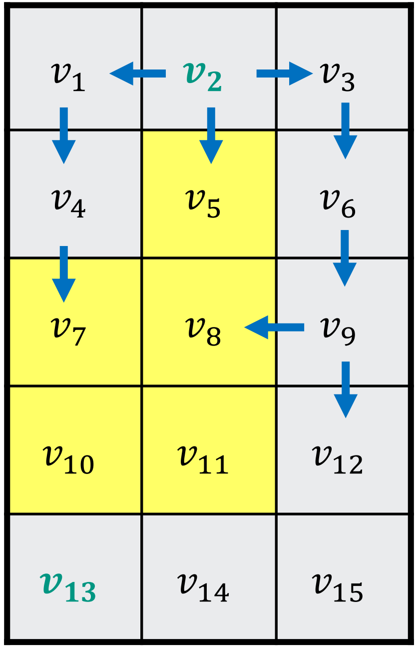
Exercise 4.1: A*- Algorithm, Step 6 [optional] (2)

- State:
  - $O = \{v_5, v_7, v_9\}$
    - $f(v_5) \approx 7.16$
    - $f(v_7) = 8$
    - $f(v_9) \approx 5.83$
  - $C = \{v_1, v_2, v_3, v_4, v_6\}$
- Update:
  - Expand $\boldsymbol{v_9}$
  - $O = \{v_5, v_7, \boldsymbol{v_8}, \boldsymbol{v_{12}}\}$
    - $f(v_8) = 3 + 4 + h(v_8) = 7 + \sqrt{1^2 + 2^2} \approx 9.24$
    - $f(v_{12}) = 3 + 1 + h(v_{12}) = 4 + \sqrt{2^2 + 1^2} \approx 6.24$
  - $C = \{v_1, v_2, v_3, v_4, v_6, \boldsymbol{v_9}\}$



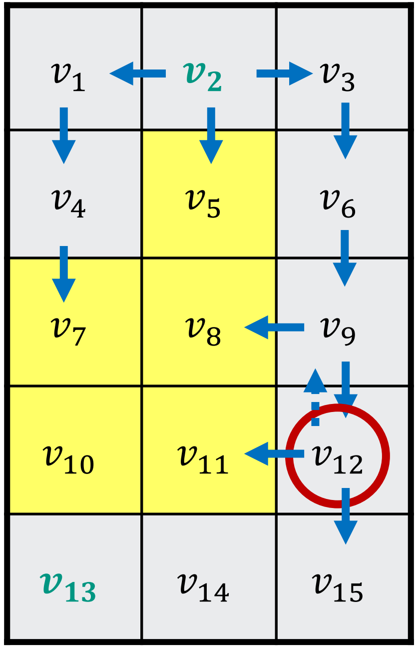
Exercise 4.1: A*- Algorithm, Step 7 [optional] (1)

- State:
  - $O = \{v_5, v_7, v_8, v_{12}\}$
    - $f(v_5) \approx 7.16$
    - $f(v_7) = 8$
    - $f(v_8) \approx 9.24$
    - $f(v_{12}) \approx 6.24$
  - $C = \{v_1, v_2, v_3, v_4, v_6, v_9\}$
- Update:



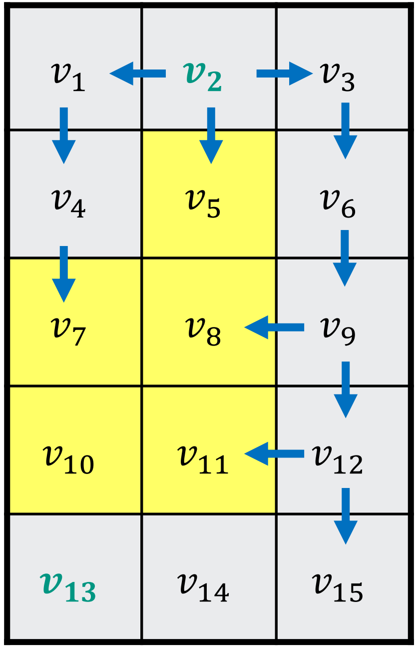
Exercise 4.1: A*- Algorithm, Step 7 [optional] (2)

- State:
  - $O = \{v_5, v_7, v_8, v_{12}\}$
    - $f(v_5) \approx 7.16$
    - $f(v_7) = 8$
    - $f(v_8) \approx 9.24$
    - $f(v_{12}) \approx 6.24$
  - $C = \{v_1, v_2, v_3, v_4, v_6, v_9\}$
- Update:
  - Expand $\boldsymbol{v_{12}}$
  - $O = \{v_5, v_7, v_8, \boldsymbol{v_{11}}, \boldsymbol{v_{15}}\}$
    - $f(v_{11}) = 4 + 4 + h(v_{11}) = 8 + \sqrt{1^2 + 1^2} \approx 9.41$
    - $f(v_{15}) = 4 + 1 + h(v_{15}) = 5 + 2 = 7$
  - $C = \{v_1, v_2, v_3, v_4, v_6, v_9, \boldsymbol{v_{12}}\}$



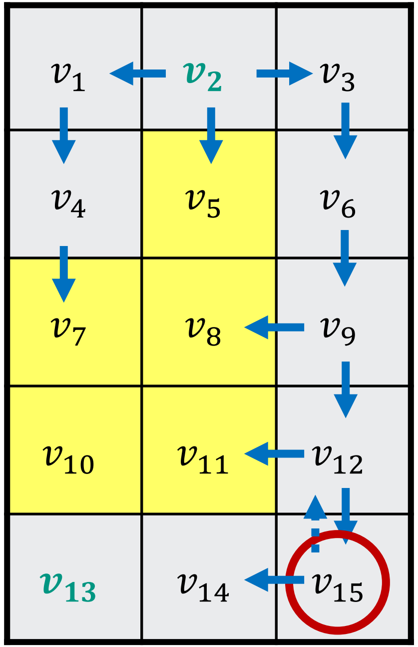
Exercise 4.1: A*- Algorithm, Step 8 [optional] (1)

- State:
  - $O = \{v_5, v_7, v_8, v_{11}, v_{15}\}$
    - $f(v_5) \approx 7.16$
    - $f(v_7) = 8$
    - $f(v_8) \approx 9.24$
    - $f(v_{11}) \approx 9.41$
    - $f(v_{15}) = 7$
  - $C = \{v_1, v_2, v_3, v_4, v_6, v_9, v_{12}\}$
- Update:



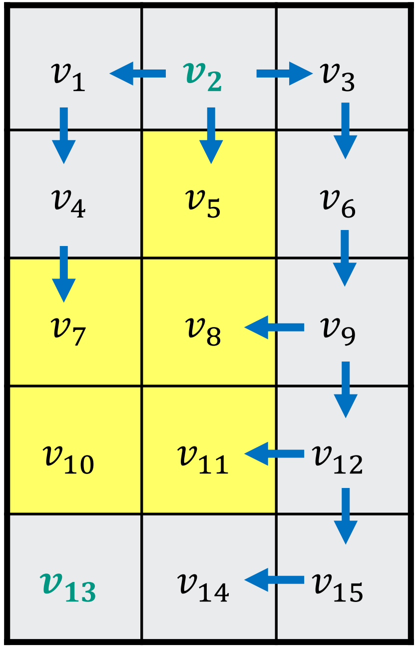
Exercise 4.1: A*- Algorithm, Step 8 [optional] (2)

- State:
  - $O = \{v_5, v_7, v_8, v_{11}, v_{15}\}$
    - $f(v_5) \approx 7.16$
    - $f(v_7) = 8$
    - $f(v_8) \approx 9.24$
    - $f(v_{11}) \approx 9.41$
    - $f(v_{15}) = 7$
  - $C = \{v_1, v_2, v_3, v_4, v_6, v_9, v_{12}\}$
- Update:
  - Expand $\boldsymbol{v_{15}}$
  - $O = \{v_5, v_7, v_8, v_{11}, \boldsymbol{v_{14}}\}$
    - $f(v_{14}) = 5 + 1 + h(v_{14}) = 6 + 1 = 7$
  - $C = \{v_1, v_2, v_3, v_4, v_6, v_9, v_{12}, \boldsymbol{v_{15}}\}$



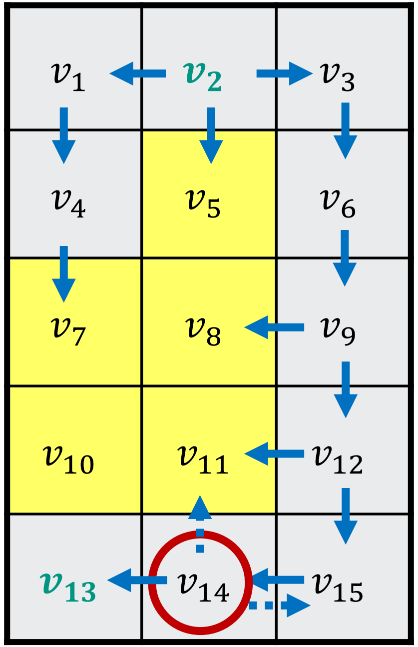
Exercise 4.1: A*- Algorithm, Step 9 [optional] (1)

- State:
  - $O = \{v_5, v_7, v_8, v_{11}, v_{14}\}$
    - $f(v_5) \approx 7.16$
    - $f(v_7) = 8$
    - $f(v_8) \approx 9.24$
    - $f(v_{11}) \approx 9.41$
    - $f(v_{14}) = 7$
  - $C = \{v_1, v_2, v_3, v_4, v_6, v_9, v_{12}, v_{15}\}$
- Update:



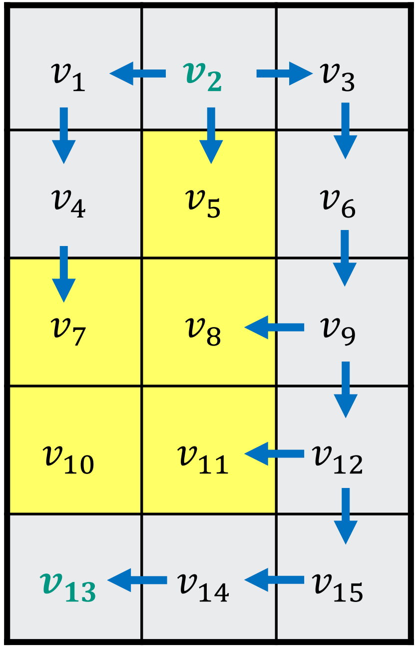
Exercise 4.1: A*- Algorithm, Step 9 [optional] (2)

- State:
  - $O = \{v_5, v_7, v_8, v_{11}, v_{14}\}$
    - $f(v_5) \approx 7.16$
    - $f(v_7) = 8$
    - $f(v_8) \approx 9.24$
    - $f(v_{11}) \approx 9.41$
    - $f(v_{14}) = 7$
  - $C = \{v_1, v_2, v_3, v_4, v_6, v_9, v_{12}, v_{15}\}$
- Update:
  - Expand $\boldsymbol{v_{14}}$
  - $O = \{v_5, v_7, v_8, \boldsymbol{v_{11}}, \boldsymbol{v_{13}}\}$
    - $f(v_{11}) = 6 + 4 + h(v_{14}) = 10 + \sqrt{2} \approx 11.41 > 9.41$
    - $f(v_{13}) = 6 + 1 + h(v_{13}) = 7 + 0 = 7$
  - $C = \{v_1, v_2, v_3, v_4, v_6, v_9, v_{12}, \boldsymbol{v_{14}}, v_{15}\}$



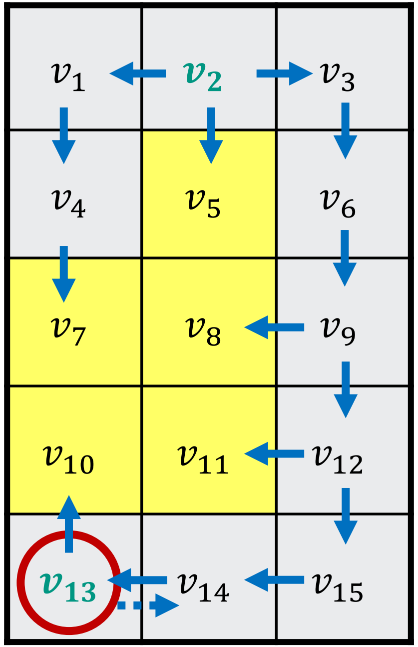
Exercise 4.1: A*- Algorithm, Step 10 [optional] (1)

- State:
  - $O = \{v_5, v_7, v_8, v_{11}, v_{13}\}$
    - $f(v_5) \approx 7.16$
    - $f(v_7) = 8$
    - $f(v_8) \approx 9.24$
    - $f(v_{11}) \approx 9.41$
    - $f(v_{13}) = 7$
  - $C = \{v_1, v_2, v_3, v_4, v_6, v_9, v_{12}, v_{14}, v_{15}\}$
- Update:



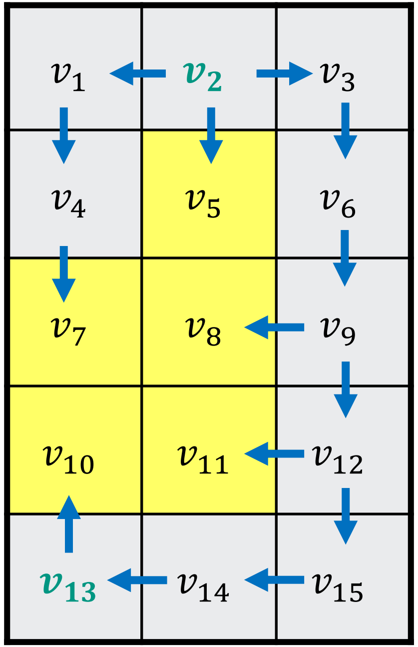
Exercise 4.1: A*- Algorithm, Step 10 [optional] (2)

- State:
  - $O = \{v_5, v_7, v_8, v_{11}, v_{13}\}$
    - $f(v_5) \approx 7.16$
    - $f(v_7) = 8$
    - $f(v_8) \approx 9.24$
    - $f(v_{11}) \approx 9.41$
    - $f(v_{13}) = 7$
  - $C = \{v_1, v_2, v_3, v_4, v_6, v_9, v_{12}, v_{14}, v_{15}\}$
- Update:
  - Expand $\boldsymbol{v_{13}}$
  - $O = \{v_5, v_7, v_8, \boldsymbol{v_{10}}, v_{11}\}$
    - $f(v_{10}) = 7 + 4 + h(v_{10}) = 11 + 1 = 12$
  - $C = \{v_1, v_2, v_3, v_4, v_6, v_9, v_{12}, \boldsymbol{v_{13}}, v_{14}, v_{15}\}$



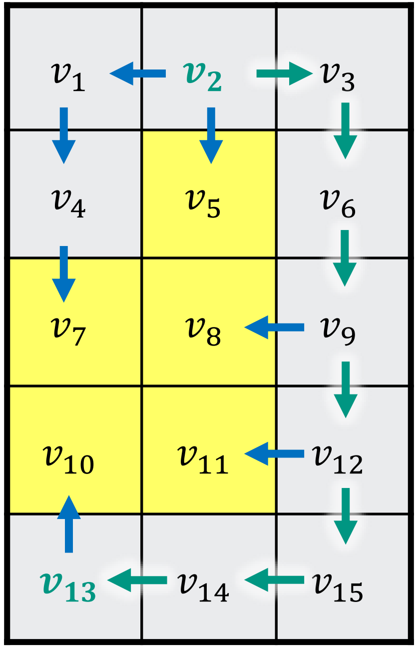
Exercise 4.1: A*- Algorithm, Step 11 [optional] (1)

- State:
  - $O = \{v_5, v_7, v_8, v_{10}, v_{11}\}$
    - $f(v_5) \approx 7.16$
    - $f(v_7) = 8$
    - $f(v_8) \approx 9.24$
    - $f(v_{10}) \approx 12$
    - $f(v_{11}) \approx 9.41$
  - $C = \{v_1, v_2, v_3, v_4, v_6, v_9, v_{12}, v_{13}, v_{14}, v_{15}\}$



Exercise 4.1: A*- Algorithm, Step 11 [optional] (2)

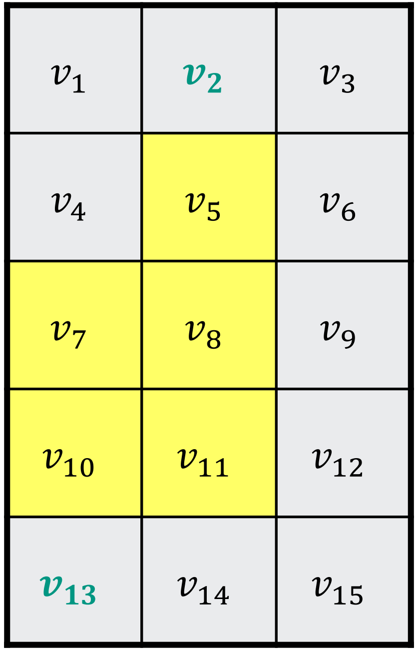
- State:
  - $O = \{v_5, v_7, v_8, v_{10}, v_{11}\}$
    - $f(v_5) \approx 7.16$
    - $f(v_7) = 8$
    - $f(v_8) \approx 9.24$
    - $f(v_{10}) \approx 12$
    - $f(v_{11}) \approx 9.41$
  - $C = \{v_1, v_2, v_3, v_4, v_6, v_9, v_{12}, v_{13}, v_{14}, v_{15}\}$
- Termination:
  - Target node v_{13} was expanded ⇒ Goal is reached
  - Traverse predecessors of v_{13} to determine a path
  - ⇒ $v_2, v_3, v_6, v_9, v_{12}, v_{15}, v_{14}, v_{13}$



Exercise 4.2: A*- Algorithm, Suitable heuristic (1)

- Only horizontal and vertical movements allowed
- Costs:
  - Entering a grey cell: 1
  - Entering a yellow cell: 4
- Heuristic h:
  Euclidean distance to v_{13}

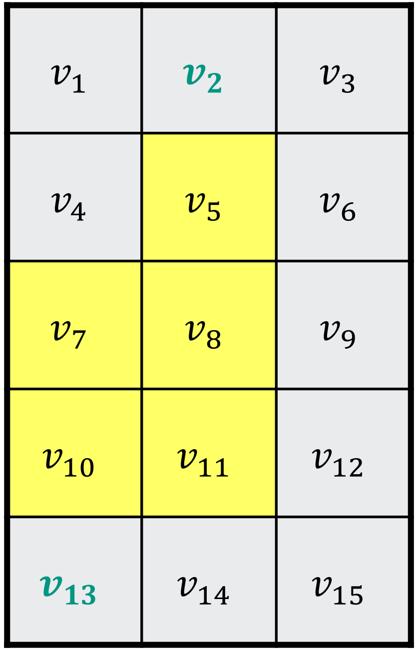
- Why is the Euclidean distance a suitable heuristic in this task?



Exercise 4.2: A*- Algorithm, Suitable heuristic (2)

- Only horizontal and vertical movements allowed
- Costs:
  - Entering a grey cell: 1
  - Entering a yellow cell: 4
- Heuristic h:
  Euclidean distance to v_{13}

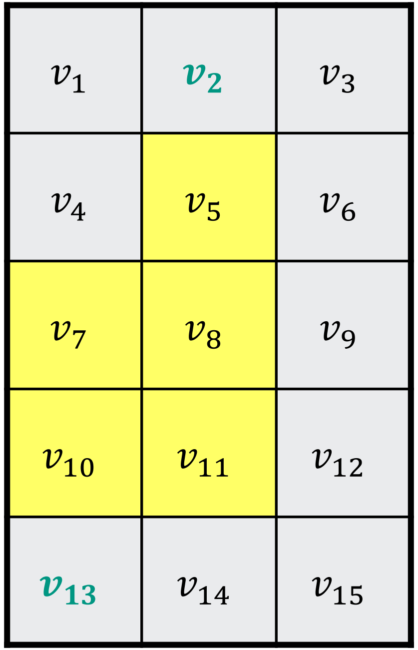
- Why is the Euclidean distance a suitable heuristic in this task?
  - Heuristic must not overestimate the remaining costs to the goal state



Exercise 4.2: A*- Algorithm, Suitable heuristic (3)

- Only horizontal and vertical movements allowed
- Costs:
  - Entering a grey cell: 1
  - Entering a yellow cell: 4
- Heuristic h:
  Euclidean distance to v_{13}

- Why is the Euclidean distance a suitable heuristic in this task?
  - Heuristic must not overestimate the remaining costs to the goal state
  - The Euclidean distance is suitable as the cost to enter a cell is always ≥ 1



Exercise 4.3: A*- Algorithm, Termination (1)

- When does the A* algorithm find a valid solution?
  a) When the next node to be expanded is the target node.
  b) When the target node is added to the Open Set.

- Justify your answer.

Exercise 4.3: A*- Algorithm, Termination (2)

- **Algorithm**
  while $O \neq \emptyset$
  - Determine next node to expand
    - find $v_i \in O$ with minimal $f(v_i) = g(v_i) + h(v_i)$
  - if $v_i = v_{goal}$
    found solution: traverse predecessor of v_i until v_{start} is reached
  - $O.remove(v_i)$
  - $C.add(v_i)$
  - **Update** all succesors v_j of v_i

    if $v_j \in C$, skip v_j

    if $v_j \notin O$, $O.add(v_j)$

    if $g(v_i) + cost(v_i, v_j) < g(v_j)$

    $$g(v_j) = g(v_i) + cost(v_i, v_j)$$
    $$h(v_j) = heuristic(v_j, v_{goal})$$
    $$pred(v_j) = v_i$$

Exercise 4.3: A*- Algorithm, Termination (3)

- **Algorithm**
  while $O \neq \emptyset$
  - Determine next node to expand
    - find $v_i \in O$ with minimal $f(v_i) = g(v_i) + h(v_i)$
(a)
  - if $v_i = v_{goal}$
    found solution: traverse predecessor of v_i until v_{start} is reached
  - $O.remove(v_i)$
  - $C.add(v_i)$
  - **Update** all succesors v_j of v_i

    if $v_j \in C$, skip v_j

(b) if $v_j \notin O$, $O.add(v_j)$

    if $g(v_i) + cost(v_i, v_j) < g(v_j)$

    $g(v_j) = g(v_i) + cost(v_i, v_j)$

    $h(v_j) = heuristic(v_j, v_{goal})$

    $pred(v_j) = v_i$

Exercise 4.3: A*- Algorithm, Termination (5)

- When does the A* algorithm find a valid solution?
  a) When the next node to be expanded is the target node.
  b) When the target node is added to the Open Set.

- Justify your answer.
  - Option (b), adding the target node to the Open Set:
    - Only one path to the target node was found
    - There may still be shorter paths to the target node
    - Algorithm cannot yet terminate
  - Option (a), expanding the target node:
    - There can be no shorter path to the target node (provided that the heuristic is **suitable**)

Exercise 4: A*- Algorithm, Bonus

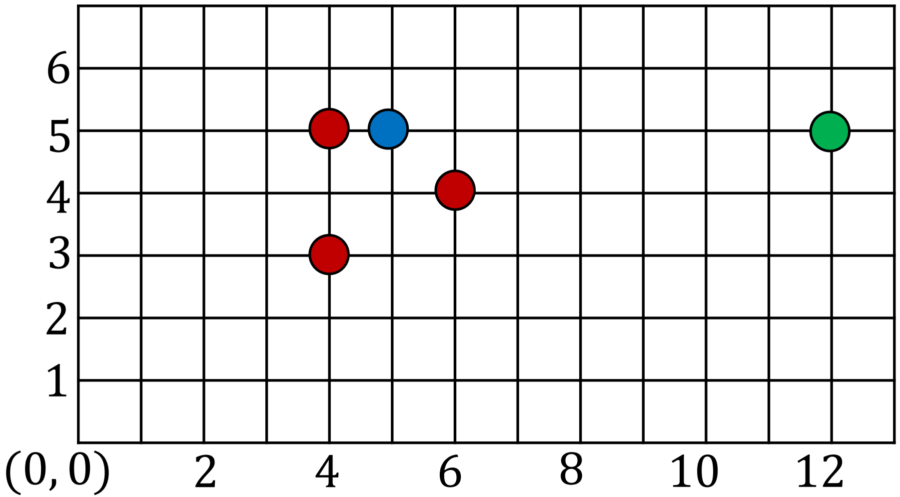
Bonus questions:

- Is the Euclidean distance a suitable heuristic if diagonal movements with equal costs are permitted?
- Is the Manhattan distance a valid heuristic for the original problem?
- If so, is the Manhattan distance a better or worse heuristic than the Euclidean distance for the original problem?

v_1	v_2	v_3
v_4	v_5	v_6
v_7	v_8	v_9
v_{10}	v_{11}	v_{12}
v_{13}	v_{14}	v_{15}

Exercise 5: Potential fields

- Robot: $\boldsymbol{q}_R$
- Obstacles: $\boldsymbol{q}_{rep,1}, \boldsymbol{q}_{rep,2}, \boldsymbol{q}_{rep,3}$
- Goal: $\boldsymbol{q}_{goal}$



1. Which repulsive potentials act on the robot with $\rho_0 = 5$?
2. Determine $U(\boldsymbol{q}_R)$ as the sum of the acting potential fields.
3. Determine the direction in which the robot would move.

$$\boldsymbol{q}_R = \binom{5}{5}$$

$$\boldsymbol{q}_{rep,1} = \binom{4}{3}, \boldsymbol{q}_{rep,2} = \binom{6}{4}, \boldsymbol{q}_{rep,3} = \binom{4}{5}$$

$$\boldsymbol{q}_{goal} = \binom{12}{5}$$

Exercise 5.1: Potential fields – Repulsive potentials (1)

- Obstacles create a repulsive potential
- The robot shall **not** be influenced for large distances to obstacles ($> \rho_0$)
- **Example (FIRAS function):**

$$U_{rep}(\boldsymbol{q}) = \begin{cases} \frac{1}{2}\nu\left(\frac{1}{\rho(\boldsymbol{q},\boldsymbol{q}_{obs})} - \frac{1}{\rho_0}\right)^2 & if \;\; \rho(\boldsymbol{q},\boldsymbol{q}_{obs}) \leq \rho_0 \\ 0 & else \end{cases}$$

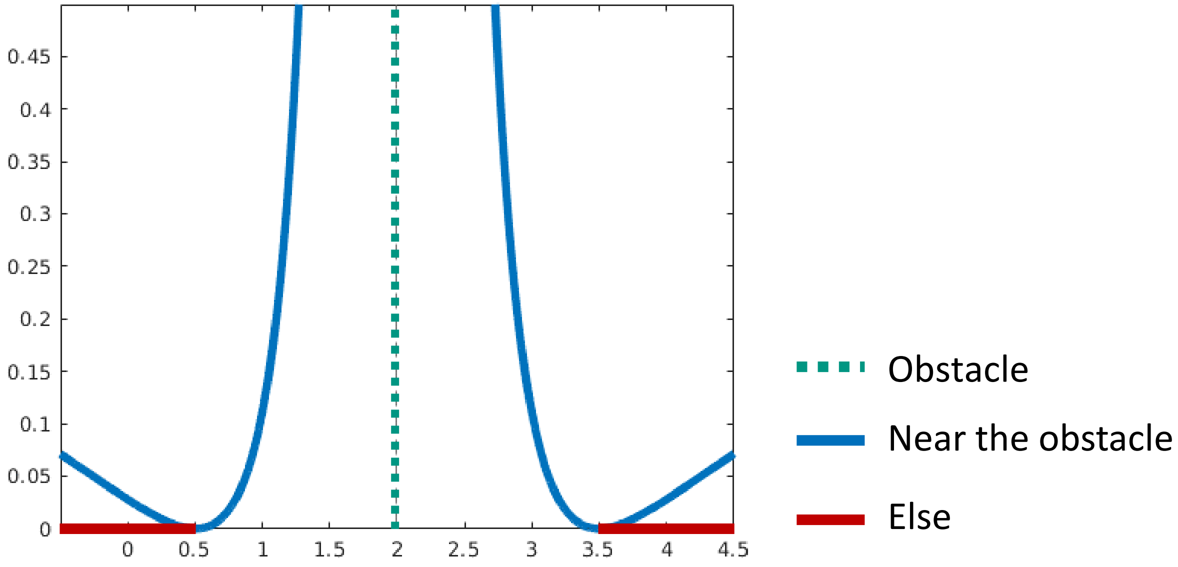
$\rho(\mathbf{q},\mathbf{q}_{obs}) = \|\boldsymbol{q} - \boldsymbol{q}_{obs}\|$ is the distance between the robot and the obstacle

$$F_{rep} = -\nabla U_{rep} = \nu\left(\frac{1}{\rho(\boldsymbol{q},\boldsymbol{q}_{obs})} - \frac{1}{\rho_0}\right) \cdot \frac{1}{\rho(\boldsymbol{q},\boldsymbol{q}_{obs})^2} \cdot \frac{\boldsymbol{q} - \boldsymbol{q}_{obs}}{\rho(\boldsymbol{q},\boldsymbol{q}_{obs})}$$

$$\frac{\partial \|\boldsymbol{x}\|}{\partial \boldsymbol{x}} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$$

Exercise 5.1: Potential fields – Repulsive potentials (2)

$$U_{rep}(x) = \begin{cases} \left(\frac{1}{\|x-2\|} - \frac{1}{1.5}\right)^2 & if\ \|x-2\| \leq 1.5 \\ 0 & else \end{cases}$$



Exercise 5.1: Potential fields – Repulsive potentials (3)

- $\boldsymbol{q}_R = \begin{pmatrix} 5 \\ 5 \end{pmatrix}$
- $\boldsymbol{q}_{rep,1} = \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \boldsymbol{q}_{rep,2} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}, \boldsymbol{q}_{rep,3} = \begin{pmatrix} 4 \\ 5 \end{pmatrix}$
- $\rho_0 = 5$

Exercise 5.1: Potential fields – Repulsive potentials (4)

- $\boldsymbol{q}_R = \begin{pmatrix} 5 \\ 5 \end{pmatrix}$
- $\boldsymbol{q}_{rep,1} = \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \boldsymbol{q}_{rep,2} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}, \boldsymbol{q}_{rep,3} = \begin{pmatrix} 4 \\ 5 \end{pmatrix}$
- $\rho_0 = 5$

$$\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,1}\| = \left\|\begin{pmatrix} 5 \\ 5 \end{pmatrix} - \begin{pmatrix} 4 \\ 3 \end{pmatrix}\right\| = \left\|\begin{pmatrix} 1 \\ 2 \end{pmatrix}\right\| = \sqrt{1^2 + 2^2} = \sqrt{5} \approx 2.2$$

$$\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,2}\| = \left\|\begin{pmatrix} 5 \\ 5 \end{pmatrix} - \begin{pmatrix} 6 \\ 4 \end{pmatrix}\right\| = \left\|\begin{pmatrix} -1 \\ 1 \end{pmatrix}\right\| = \sqrt{(-1)^2 + 1^2} = \sqrt{2} \approx 1.4$$

$$\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,3}\| = \left\|\begin{pmatrix} 5 \\ 5 \end{pmatrix} - \begin{pmatrix} 4 \\ 5 \end{pmatrix}\right\| = \left\|\begin{pmatrix} 1 \\ 0 \end{pmatrix}\right\| = \sqrt{1^2 + 0^2} = \sqrt{1} = 1$$

Exercise 5.1: Potential fields – Repulsive potentials (5)

- $\boldsymbol{q}_R = \begin{pmatrix} 5 \\ 5 \end{pmatrix}$
- $\boldsymbol{q}_{rep,1} = \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \boldsymbol{q}_{rep,2} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}, \boldsymbol{q}_{rep,3} = \begin{pmatrix} 4 \\ 5 \end{pmatrix}$
- $\rho_0 = 5$

$$\left\| \boldsymbol{q}_R - \boldsymbol{q}_{rep,1} \right\| \approx 2.2 < \rho_0$$
$$\left\| \boldsymbol{q}_R - \boldsymbol{q}_{rep,2} \right\| \approx 1.4 < \rho_0$$
$$\left\| \boldsymbol{q}_R - \boldsymbol{q}_{rep,3} \right\| = 1 < \rho_0$$

➔ All three repulsive potentials act on the robot.

Potential Fields – Attracting potential

- **Attracting potential**

  → There shall be only a single minimum, located at $\boldsymbol{q}_{goal}$

- **Linear function** of the distance to the goal:

$$U_{attr}(\boldsymbol{q}) = k \cdot \|\boldsymbol{q} - \boldsymbol{q}_{goal}\|$$

$$F_{\boldsymbol{attr}}(\boldsymbol{q}) = -\nabla U_{attr}(\boldsymbol{q}) = -k \cdot \frac{\boldsymbol{q} - \boldsymbol{q}_{goal}}{\|\boldsymbol{q} - \boldsymbol{q}_{\boldsymbol{goal}}\|}$$

$$\frac{\partial \|\boldsymbol{x}\|}{\partial \boldsymbol{x}} = \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}$$

Exercise 5.2: Sum of the acting potential fields (1)

Karlsruher Institut für Technologie

$$U(\boldsymbol{q}_R) = U_{attr}(\boldsymbol{q}_R) + \sum_{i=1}^{3} U_{rep,i}(\boldsymbol{q}_R) \qquad \text{with } k = 1, v = 1, \rho_0 = 5$$

Exercise 5.2: Sum of the acting potential fields (2)

$$U(\boldsymbol{q}_R) = U_{attr}(\boldsymbol{q}_R) + \sum_{i=1}^{3} U_{rep,i}(\boldsymbol{q}_R) \qquad \text{with } k = 1, v = 1, \rho_0 = 5$$

$$U_{attr}(\boldsymbol{q}_R) = k \cdot \left\| \boldsymbol{q}_R - \boldsymbol{q}_{goal} \right\| = \left\| \boldsymbol{q}_R - \boldsymbol{q}_{goal} \right\|$$

Exercise 5.2: Sum of the acting potential fields (3)

$$U(\boldsymbol{q}_R) = U_{attr}(\boldsymbol{q}_R) + \sum_{i=1}^{3} U_{rep,i}(\boldsymbol{q}_R) \qquad \text{with } k = 1, v = 1, \rho_0 = 5$$

$$U_{attr}(\boldsymbol{q}_R) = k \cdot \left\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\right\| = \left\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\right\|$$

$$U_{rep,i}(\boldsymbol{q}_R) = \frac{1}{2} v \left(\frac{1}{\left\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\right\|} - \frac{1}{\rho_0}\right)^2 = \frac{1}{2} \left(\frac{1}{\left\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\right\|} - \frac{1}{\rho_0}\right)^2$$

Exercise 5.2: Sum of the acting potential fields (4)

$$U(\boldsymbol{q}_R) = U_{attr}(\boldsymbol{q}_R) + \sum_{i=1}^{3} U_{rep,i}(\boldsymbol{q}_R) \qquad \text{with } k = 1, v = 1, \rho_0 = 5$$

$$U_{attr}(\boldsymbol{q}_R) = k \cdot \left\| \boldsymbol{q}_R - \boldsymbol{q}_{goal} \right\| = \left\| \boldsymbol{q}_R - \boldsymbol{q}_{goal} \right\|$$

$$U_{rep,i}(\boldsymbol{q}_R) = \frac{1}{2} v \left(\frac{1}{\left\| \boldsymbol{q}_R - \boldsymbol{q}_{rep,i} \right\|} - \frac{1}{\rho_0} \right)^2 = \frac{1}{2} \left(\frac{1}{\left\| \boldsymbol{q}_R - \boldsymbol{q}_{rep,i} \right\|} - \frac{1}{\rho_0} \right)^2$$

$$U(\boldsymbol{q}_R) = \left\| \boldsymbol{q}_R - \boldsymbol{q}_{goal} \right\| + \sum_{i=1}^{3} \frac{1}{2} \left(\frac{1}{\left\| \boldsymbol{q}_R - \boldsymbol{q}_{rep,i} \right\|} - \frac{1}{\rho_0} \right)^2$$

Exercise 5.3: Direction of the robot (1)

Direction: Which force acts on the robot?

$$F(\boldsymbol{q}_R) = -\nabla U(\boldsymbol{q}_R)$$

$$= -\nabla \left(U_{attr}(\boldsymbol{q}_R) + \sum_{i=1}^{3} U_{rep,i}(\boldsymbol{q}_R) \right)$$

$$= -\nabla U_{attr}(\boldsymbol{q}_R) + \sum_{i=1}^{3} -\nabla U_{rep,i}(\boldsymbol{q}_R)$$

Exercise 5.3: Direction of the robot (2)

Richtung: Welche Kraft wirkt auf den Roboter?

$$F(\boldsymbol{q}_R) = -\nabla U(\boldsymbol{q}_R)$$

$$= -\nabla \left(U_{attr}(\boldsymbol{q}_R) + \sum_{i=1}^{3} U_{rep,i}(\boldsymbol{q}_R) \right)$$

$$= -\nabla U_{attr}(\boldsymbol{q}_R) + \sum_{i=1}^{3} -\nabla U_{rep,i}(\boldsymbol{q}_R)$$

$$= -\frac{\boldsymbol{q}_R - \boldsymbol{q}_{goal}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\|} + \sum_{i=1}^{3} \left(\frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|} - \frac{1}{\rho_0} \right) \cdot \frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|^2} \cdot \frac{\boldsymbol{q}_{\boldsymbol{R}} - \boldsymbol{q}_{rep,i}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|}$$

Exercise 5.3: Direction of the robot (4)

$$F(\boldsymbol{q}_R) = -\frac{\boldsymbol{q}_R - \boldsymbol{q}_{goal}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\|} + \sum_{i=1}^{3} \left(\frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|} - \frac{1}{\rho_0} \right) \cdot \frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|^2} \cdot \frac{\boldsymbol{q}_{\boldsymbol{R}} - \boldsymbol{q}_{rep,i}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|}$$

$$\boldsymbol{q}_R = \begin{pmatrix} 5 \\ 5 \end{pmatrix}, \boldsymbol{q}_{rep,1} = \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \boldsymbol{q}_{rep,2} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}, \boldsymbol{q}_{rep,3} = \begin{pmatrix} 4 \\ 5 \end{pmatrix}, \boldsymbol{q}_{goal} = \begin{pmatrix} 12 \\ 5 \end{pmatrix}, \rho_0 = 5$$

Exercise 5.3: Direction of the robot (5)

$$F(\boldsymbol{q}_R) = -\frac{\boldsymbol{q}_R - \boldsymbol{q}_{goal}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\|} + \sum_{i=1}^{3}\left(\frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|} - \frac{1}{\rho_0}\right) \cdot \frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|^2} \cdot \frac{\boldsymbol{q}_{\boldsymbol{R}} - \boldsymbol{q}_{rep,i}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|}$$

$$\boldsymbol{q}_R = \begin{pmatrix}5\\5\end{pmatrix}, \boldsymbol{q}_{rep,1} = \begin{pmatrix}4\\3\end{pmatrix}, \boldsymbol{q}_{rep,2} = \begin{pmatrix}6\\4\end{pmatrix}, \boldsymbol{q}_{rep,3} = \begin{pmatrix}4\\5\end{pmatrix}, \boldsymbol{q}_{goal} = \begin{pmatrix}12\\5\end{pmatrix}, \rho_0 = 5$$

$$\boldsymbol{q}_R - \boldsymbol{q}_{goal} = \begin{pmatrix}5\\5\end{pmatrix} - \begin{pmatrix}12\\5\end{pmatrix} = \begin{pmatrix}7\\0\end{pmatrix} \qquad \|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\| = 7$$

$$\boldsymbol{q}_R - \boldsymbol{q}_{rep,1} = \begin{pmatrix}5\\5\end{pmatrix} - \begin{pmatrix}4\\3\end{pmatrix} = \begin{pmatrix}1\\2\end{pmatrix} \qquad \|\boldsymbol{q}_R - \boldsymbol{q}_{rep,1}\| = \sqrt{5}$$

$$\boldsymbol{q}_R - \boldsymbol{q}_{rep,2} = \begin{pmatrix}5\\5\end{pmatrix} - \begin{pmatrix}6\\4\end{pmatrix} = \begin{pmatrix}-1\\1\end{pmatrix} \qquad \|\boldsymbol{q}_R - \boldsymbol{q}_{rep,2}\| = \sqrt{2}$$

$$\boldsymbol{q}_R - \boldsymbol{q}_{rep,3} = \begin{pmatrix}5\\5\end{pmatrix} - \begin{pmatrix}4\\5\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix} \qquad \|\boldsymbol{q}_R - \boldsymbol{q}_{rep,3}\| = 1$$

Exercise 5.3: Direction of the robot (6)

$$F(\boldsymbol{q}_R) = -\frac{\boldsymbol{q}_R - \boldsymbol{q}_{goal}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\|} + \sum_{i=1}^{3}\left(\frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|} - \frac{1}{5}\right) \cdot \frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|^2} \cdot \frac{\boldsymbol{q}_{\boldsymbol{R}} - \boldsymbol{q}_{rep,i}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|}$$

$\boldsymbol{q}_R - \boldsymbol{q}_{goal} = \begin{pmatrix} 7 \\ 0 \end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\| = 7$, $\boldsymbol{q}_R - \boldsymbol{q}_{rep,1} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,1}\| = \sqrt{5}$

$\boldsymbol{q}_R - \boldsymbol{q}_{rep,2} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,2}\| = \sqrt{2}$, $\boldsymbol{q}_R - \boldsymbol{q}_{rep,3} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,3}\| = 1$

$$F(\boldsymbol{q}_R) =$$

Exercise 5.3: Direction of the robot (7)

$$F(\boldsymbol{q}_R) = -\frac{\boldsymbol{q}_R - \boldsymbol{q}_{goal}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\|} + \sum_{i=1}^{3}\left(\frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|} - \frac{1}{5}\right)\cdot\frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|^2}\cdot\frac{\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|}$$

$\boldsymbol{q}_R - \boldsymbol{q}_{goal} = \begin{pmatrix}-7\\0\end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\| = 7$, $\boldsymbol{q}_R - \boldsymbol{q}_{rep,1} = \begin{pmatrix}1\\2\end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,1}\| = \sqrt{5}$

$\boldsymbol{q}_R - \boldsymbol{q}_{rep,2} = \begin{pmatrix}-1\\1\end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,2}\| = \sqrt{2}$, $\boldsymbol{q}_R - \boldsymbol{q}_{rep,3} = \begin{pmatrix}1\\0\end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,3}\| = 1$

$$F(\boldsymbol{q}_R) = -\begin{pmatrix}-1\\0\end{pmatrix} +$$

Exercise 5.3: Direction of the robot (8)

$$F(\boldsymbol{q}_R) = -\frac{\boldsymbol{q}_R - \boldsymbol{q}_{goal}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\|} + \sum_{i=1}^{3} \left(\frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|} - \frac{1}{5} \right) \cdot \frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|^2} \cdot \frac{\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|}$$

$\boldsymbol{q}_R - \boldsymbol{q}_{goal} = \begin{pmatrix} -7 \\ 0 \end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\| = 7$, $\boldsymbol{q}_R - \boldsymbol{q}_{rep,1} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,1}\| = \sqrt{5}$

$\boldsymbol{q}_R - \boldsymbol{q}_{rep,2} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,2}\| = \sqrt{2}$, $\boldsymbol{q}_R - \boldsymbol{q}_{rep,3} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,3}\| = 1$

$$F(\boldsymbol{q}_R) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \left(\frac{1}{\sqrt{5}} - \frac{1}{5} \right) \cdot \frac{1}{5} \cdot \frac{1}{\sqrt{5}} \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Exercise 5.3: Direction of the robot (9)

$$F(\boldsymbol{q}_R) = -\frac{\boldsymbol{q}_R - \boldsymbol{q}_{goal}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\|} + \sum_{i=1}^{3}\left(\frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|} - \frac{1}{5}\right) \cdot \frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|^2} \cdot \frac{\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|}$$

$\boldsymbol{q}_R - \boldsymbol{q}_{goal} = \begin{pmatrix}-7\\0\end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\| = 7$, $\boldsymbol{q}_R - \boldsymbol{q}_{rep,1} = \begin{pmatrix}1\\2\end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,1}\| = \sqrt{5}$

$\boldsymbol{q}_R - \boldsymbol{q}_{rep,2} = \begin{pmatrix}-1\\1\end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,2}\| = \sqrt{2}$, $\boldsymbol{q}_R - \boldsymbol{q}_{rep,3} = \begin{pmatrix}1\\0\end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,3}\| = 1$

$$F(\boldsymbol{q}_R) = \begin{pmatrix}1\\0\end{pmatrix} + \frac{5-\sqrt{5}}{125} \cdot \begin{pmatrix}1\\2\end{pmatrix} +$$

Exercise 5.3: Direction of the robot (10)

$$F(\boldsymbol{q}_R) = -\frac{\boldsymbol{q}_R - \boldsymbol{q}_{goal}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\|} + \sum_{i=1}^{3}\left(\frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|} - \frac{1}{5}\right)\cdot\frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|^2}\cdot\frac{\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|}$$

$\boldsymbol{q}_R - \boldsymbol{q}_{goal} = \begin{pmatrix}-7\\0\end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\| = 7$, $\boldsymbol{q}_R - \boldsymbol{q}_{rep,1} = \begin{pmatrix}1\\2\end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,1}\| = \sqrt{5}$

$\boldsymbol{q}_R - \boldsymbol{q}_{rep,2} = \begin{pmatrix}-1\\1\end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,2}\| = \sqrt{2}$, $\boldsymbol{q}_R - \boldsymbol{q}_{rep,3} = \begin{pmatrix}1\\0\end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,3}\| = 1$

$$F(\boldsymbol{q}_R) = \begin{pmatrix}1\\0\end{pmatrix} + \frac{5-\sqrt{5}}{125}\cdot\begin{pmatrix}1\\2\end{pmatrix} + \left(\frac{1}{\sqrt{2}} - \frac{1}{5}\right)\cdot\frac{1}{2}\cdot\frac{1}{\sqrt{2}}\begin{pmatrix}-1\\1\end{pmatrix}$$

Exercise 5.3: Direction of the robot (11)

Karlsruher Institut für Technologie

$$F(\boldsymbol{q}_R) = -\frac{\boldsymbol{q}_R - \boldsymbol{q}_{goal}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\|} + \sum_{i=1}^{3} \left(\frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|} - \frac{1}{5} \right) \cdot \frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|^2} \cdot \frac{\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|}$$

$\boldsymbol{q}_R - \boldsymbol{q}_{goal} = \begin{pmatrix} -7 \\ 0 \end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\| = 7$, $\boldsymbol{q}_R - \boldsymbol{q}_{rep,1} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,1}\| = \sqrt{5}$

$\boldsymbol{q}_R - \boldsymbol{q}_{rep,2} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,2}\| = \sqrt{2}$, $\boldsymbol{q}_R - \boldsymbol{q}_{rep,3} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,3}\| = 1$

$$F(\boldsymbol{q}_R) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{5-\sqrt{5}}{125} \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \frac{10-2\sqrt{2}}{40} \cdot \begin{pmatrix} -1 \\ 1 \end{pmatrix} +$$

Exercise 5.3: Direction of the robot (12)

$$F(\boldsymbol{q}_R) = -\frac{\boldsymbol{q}_R - \boldsymbol{q}_{goal}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\|} + \sum_{i=1}^{3} \left(\frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|} - \frac{1}{5} \right) \cdot \frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|^2} \cdot \frac{\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|}$$

$\boldsymbol{q}_R - \boldsymbol{q}_{goal} = \begin{pmatrix} -7 \\ 0 \end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\| = 7$, $\boldsymbol{q}_R - \boldsymbol{q}_{rep,1} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,1}\| = \sqrt{5}$

$\boldsymbol{q}_R - \boldsymbol{q}_{rep,2} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,2}\| = \sqrt{2}$, $\boldsymbol{q}_R - \boldsymbol{q}_{rep,3} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,3}\| = 1$

$$F(\boldsymbol{q}_R) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{5-\sqrt{5}}{125} \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \frac{10-2\sqrt{2}}{40} \cdot \begin{pmatrix} -1 \\ 1 \end{pmatrix} + \left(\frac{1}{1} - \frac{1}{5} \right) \cdot \frac{1}{1} \cdot \frac{1}{1} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Exercise 5.3: Direction of the robot (13)

$$F(\boldsymbol{q}_R) = -\frac{\boldsymbol{q}_R - \boldsymbol{q}_{goal}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\|} + \sum_{i=1}^{3}\left(\frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|} - \frac{1}{5}\right) \cdot \frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|^2} \cdot \frac{\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|}$$

$$\boldsymbol{q}_R - \boldsymbol{q}_{goal} = \begin{pmatrix}-7\\0\end{pmatrix}, \|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\| = 7, \ \boldsymbol{q}_R - \boldsymbol{q}_{rep,1} = \begin{pmatrix}1\\2\end{pmatrix}, \|\boldsymbol{q}_R - \boldsymbol{q}_{rep,1}\| = \sqrt{5}$$

$$\boldsymbol{q}_R - \boldsymbol{q}_{rep,2} = \begin{pmatrix}-1\\1\end{pmatrix}, \|\boldsymbol{q}_R - \boldsymbol{q}_{rep,2}\| = \sqrt{2}, \boldsymbol{q}_R - \boldsymbol{q}_{rep,3} = \begin{pmatrix}1\\0\end{pmatrix}, \|\boldsymbol{q}_R - \boldsymbol{q}_{rep,3}\| = 1$$

$$F(\boldsymbol{q}_R) = \begin{pmatrix}1\\0\end{pmatrix} + \frac{5-\sqrt{5}}{125} \cdot \begin{pmatrix}1\\2\end{pmatrix} + \frac{10-2\sqrt{2}}{40} \cdot \begin{pmatrix}-1\\1\end{pmatrix} + \frac{4}{5} \cdot \begin{pmatrix}1\\0\end{pmatrix}$$

Exercise 5.3: Direction of the robot (14)

$$F(\boldsymbol{q}_R) = -\frac{\boldsymbol{q}_R - \boldsymbol{q}_{goal}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\|} + \sum_{i=1}^{3}\left(\frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|} - \frac{1}{5}\right) \cdot \frac{1}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|^2} \cdot \frac{\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}}{\|\boldsymbol{q}_R - \boldsymbol{q}_{rep,i}\|}$$

$$\boldsymbol{q}_R - \boldsymbol{q}_{goal} = \begin{pmatrix}-7\\0\end{pmatrix}, \|\boldsymbol{q}_R - \boldsymbol{q}_{goal}\| = 7,\ \boldsymbol{q}_R - \boldsymbol{q}_{rep,1} = \begin{pmatrix}1\\2\end{pmatrix}, \|\boldsymbol{q}_R - \boldsymbol{q}_{rep,1}\| = \sqrt{5}$$

$$\boldsymbol{q}_R - \boldsymbol{q}_{rep,2} = \begin{pmatrix}-1\\1\end{pmatrix}, \|\boldsymbol{q}_R - \boldsymbol{q}_{rep,2}\| = \sqrt{2}, \boldsymbol{q}_R - \boldsymbol{q}_{rep,3} = \begin{pmatrix}1\\0\end{pmatrix}, \|\boldsymbol{q}_R - \boldsymbol{q}_{rep,3}\| = 1$$

$$F(\boldsymbol{q}_R) = \begin{pmatrix}1\\0\end{pmatrix} + \frac{5-\sqrt{5}}{125} \cdot \begin{pmatrix}1\\2\end{pmatrix} + \frac{10-2\sqrt{2}}{40} \cdot \begin{pmatrix}-1\\1\end{pmatrix} + \frac{4}{5} \cdot \begin{pmatrix}1\\0\end{pmatrix} \approx \begin{pmatrix}1.643\\0.224\end{pmatrix}$$

Direction: $\frac{F(\boldsymbol{q}_R)}{0.224} \approx \begin{pmatrix}7.3\\1\end{pmatrix}$

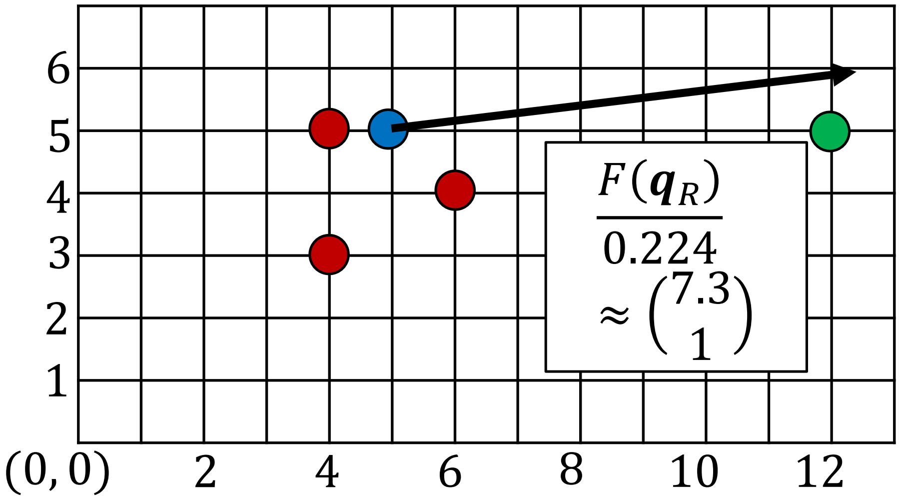
Exercise 5.3: Direction of the robot (15)

- Robot: $\boldsymbol{q}_R$
- Obstacles:
  $\boldsymbol{q}_{rep,1}, \boldsymbol{q}_{rep,2}, \boldsymbol{q}_{rep,3}$
- Goal: $\boldsymbol{q}_{goal}$

1. Which repulsive potentials act on the robot with $\rho_0 = 5$?

2. Determine $U(\boldsymbol{q}_R)$ as the sum of the acting potential fields.

3. Determine the direction in which the robot would move.



$\boldsymbol{q}_R = \begin{pmatrix} 5 \\ 5 \end{pmatrix}$

$\boldsymbol{q}_{rep,1} = \begin{pmatrix} 4 \\ 3 \end{pmatrix}, \boldsymbol{q}_{rep,2} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}, \boldsymbol{q}_{rep,3} = \begin{pmatrix} 4 \\ 5 \end{pmatrix}$

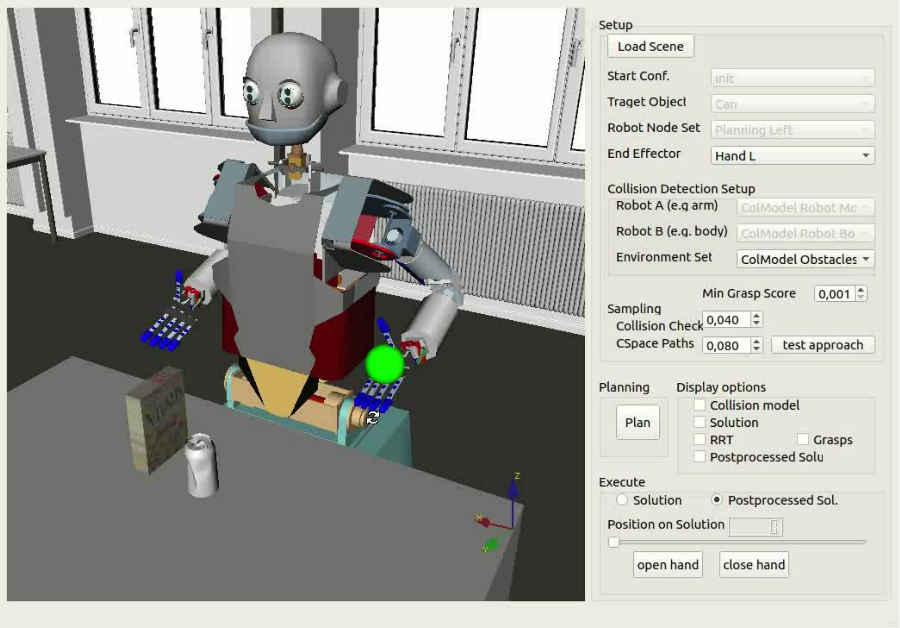
$\boldsymbol{q}_{goal} = \begin{pmatrix} 12 \\ 5 \end{pmatrix}$

Bonus: RRT in Simox (1)

Simox is a software toolbox for …

- Modeling robots, objects, scenes
- Grasp planning & motion planning
- Kinematic & physical simulation
- …

- Hier: Plan a motion using
  RRT on ARMAR-III

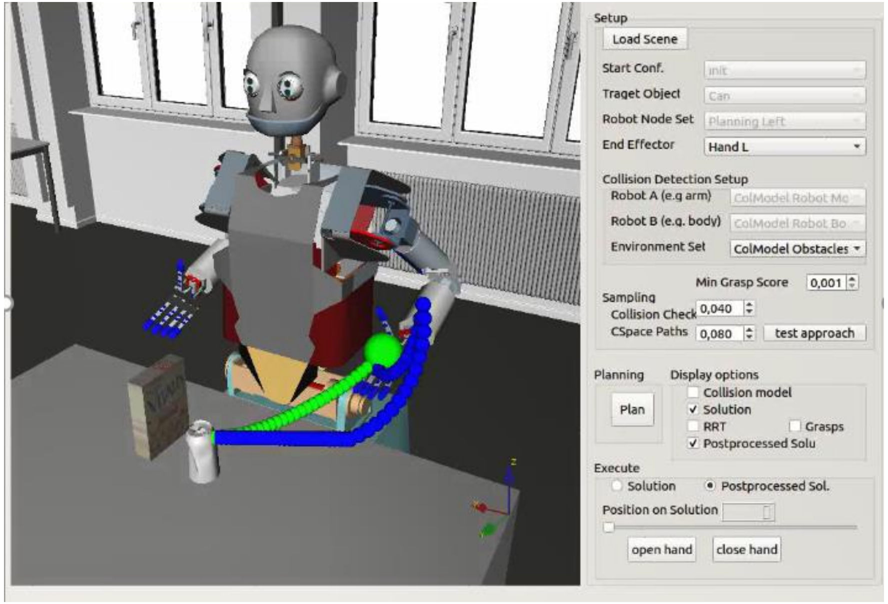


Bonus: RRT in Simox (2)

Simox is a software toolbox for …

- Modeling robots, objects, scenes
- Grasp planning & motion planning
- Kinematic & physical simulation
- …

- Hier: Plan a motion using **RRT on ARMAR-III**



Bonus: RRT in Simox: Code (1)

```
void GraspRrtWindow::plan()
{
    if (not (robot and rns and eef and graspQuality))
    {
        return;
    }

    // Setup collision detection.
    CDManagerPtr cdm(new CDManager());

    if (colModelRobA)
    {
        cdm->addCollisionModel(colModelRobA);
    }
    if (colModelRobB)
    {
        cdm->addCollisionModel(colModelRobB);
    }
    if (colModelEnv)
    {
        cdm->addCollisionModel(colModelEnv);
    }

    cdm->addCollisionModel(targetObject);
```

Setup collision detection

Register collision models

Bonus: RRT in Simox: Code (2)

```
void GraspRrtWindow::plan()
{
    // ...
```

```
unsigned int maxConfigs = 500000;
cspace = std::make_shared<Saba::CSpaceSampled>(robot, cdm, rns, maxConfigs);

float sampl = static_cast<float>(UI.doubleSpinBoxCSpaceSampling->value());
float samplDCD = static_cast<float>(UI.doubleSpinBoxColChecking->value());
float minGraspScore = static_cast<float>(UI.doubleSpinBoxMinGraspScore->value());
cspace->setSamplingSize(sampl);
cspace->setSamplingSizeDCD(samplDCD);
```

Get parameters from GUI

```
Saba::GraspRrtPtr graspRrt = std::make_shared<Saba::GraspRrt>(
            cspace, eef, targetObject, graspQuality, colModelEnv, 0.1f, minGraspScore);

graspRrt->setStart(startConfig);

bool planOK = graspRrt->plan();
```

Setup planner,
Set start config,
Start planning

https://git.h2t.iar.kit.edu/sw/simox/simox/-/blob/master/MotionPlanning/examples/GraspRRT/GraspRrtWindow.cpp#L816

Bonus: RRT in Simox: Code (3)

Karlsruher Institut für Technologie

```
void GraspRrtWindow::plan()
{
    // ...
    bool planOK = graspRrt->plan();
    if (planOK)
    {
        VR_INFO << "Planning succeeded " << std::endl;
        solution = graspRrt->getSolution();

        Saba::ShortcutProcessorPtr postProcessing =
                std::make_shared<Saba::ShortcutProcessor>(solution, cspace, false);
        int steps = 100;
        solutionOptimized = postProcessing->optimize(steps);

        tree = graspRrt->getTree();
    }
    else
    {
        VR_INFO << " Planning failed" << std::endl;
    }
    sliderSolution(1000);
    buildVisu();
}
```

- Planning
- Get solution
- Postprocess solution (smooting)
- Error handling
- Update GUI

https://git.h2t.iar.kit.edu/sw/simox/simox/-/blob/master/MotionPlanning/examples/GraspRRT/GraspRrtWindow.cpp#L816

Motion Planning: Problem Classes (1)

- **Class a**

  Known: complete world model
  complete set of constraints

  Required: collision-free trajectory from start to goal state

- **Class b**

  Known: incomplete world model
  incomplete set of constraints

  Required: collision-free trajectory from start to goal state

  Problem: collision with unknown objects

Motion Planning: Problem Classes (2)

- **Class c**
  Known: time-variant world model (moving obstacles)
  Required: collision-free trajectory from start to goal state
  Problem: changing obstacles in time and space

- **Class d**
  Known: time-variant world model
  Required: trajectory to moving goal (**rendezvous problem**)
  Problem: changing goal state in time and space

- **Class e**
  Known: no world model
  Required: collision-free trajectory from start to goal state
  Problem: Mapping (creation of world model)