4-7 샘플링 기법 5:
RRT Star (RRT*)

강의 요약

01

RRT

- Tree
- Single-query
- Probabilistic Completeness
- Narrow Passage

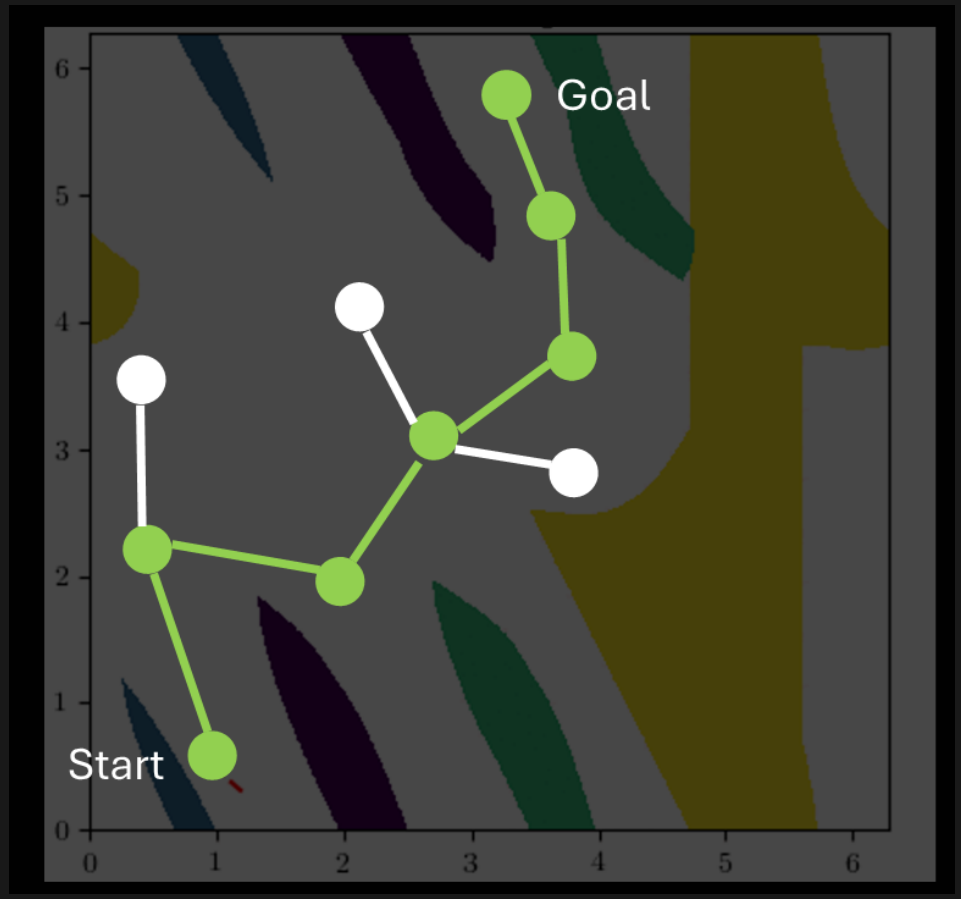
02

알고리즘

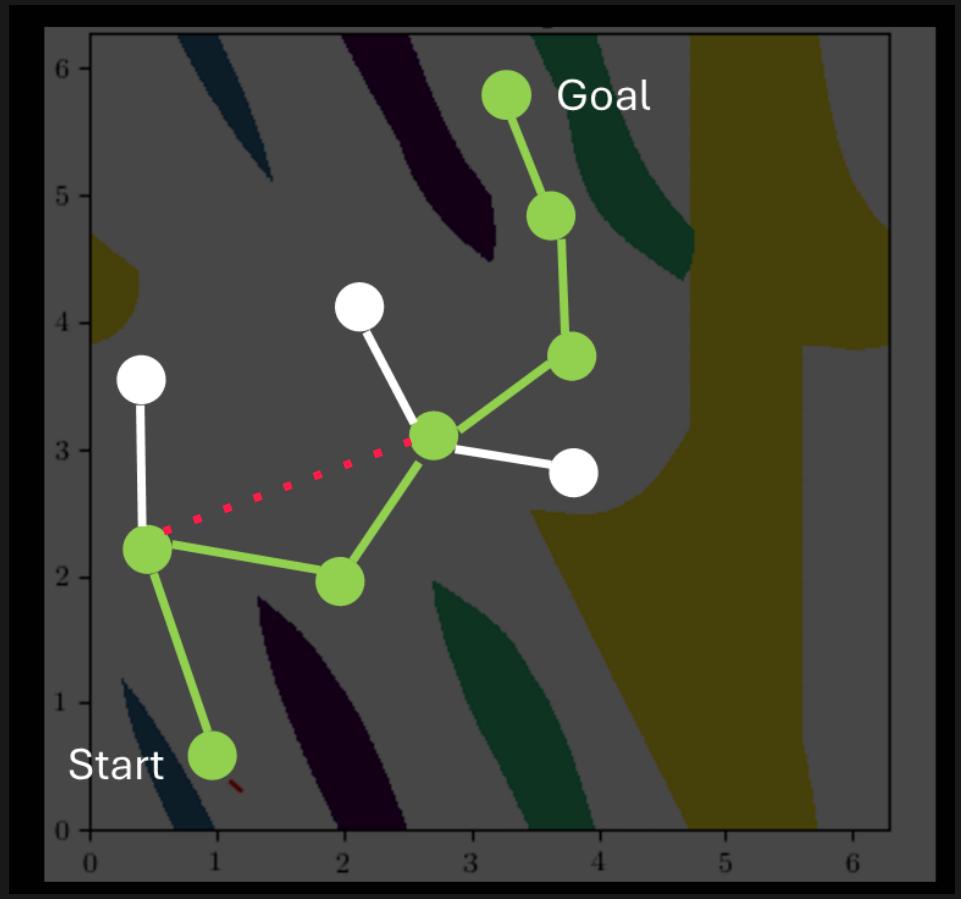
03

코드 분석

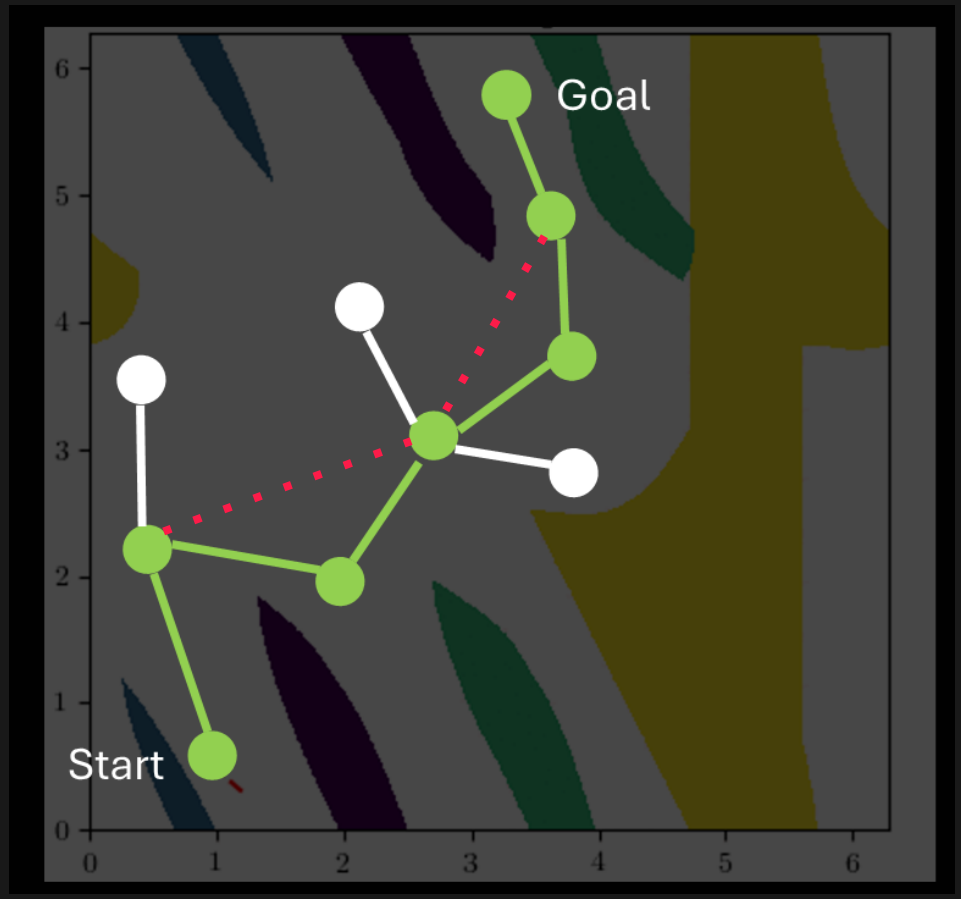
RRT 의 단점



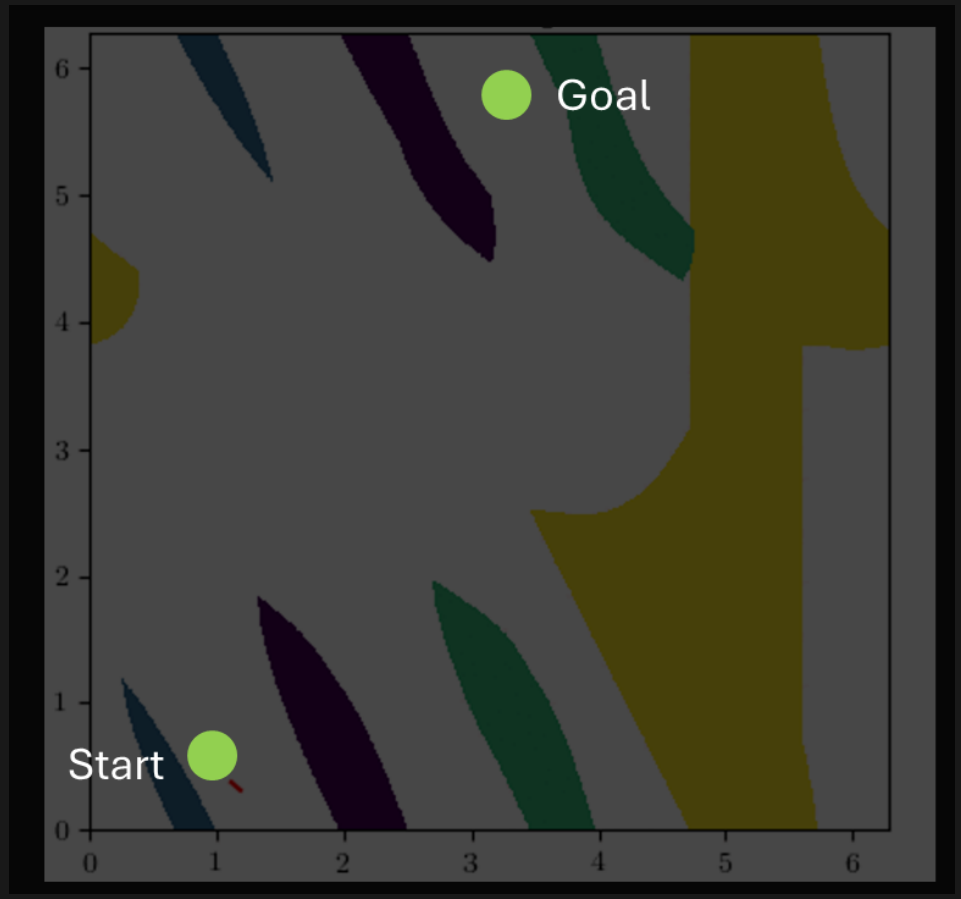
RRT 의 단점



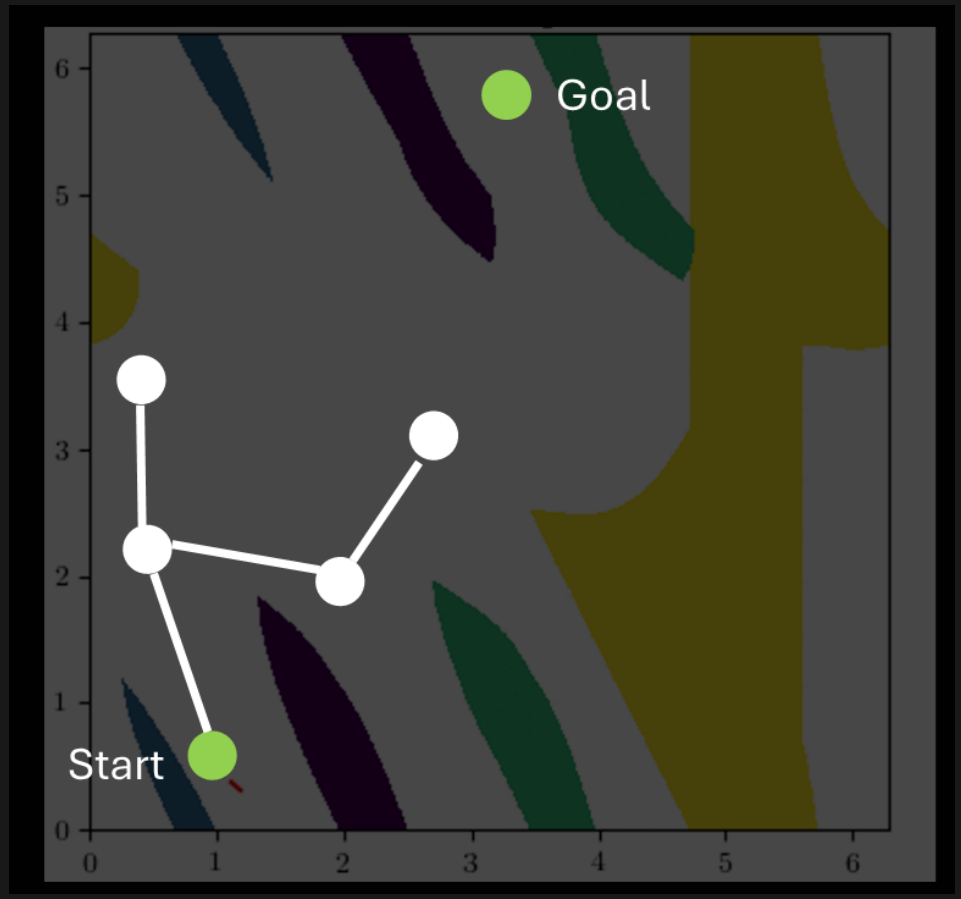
RRT 의 단점



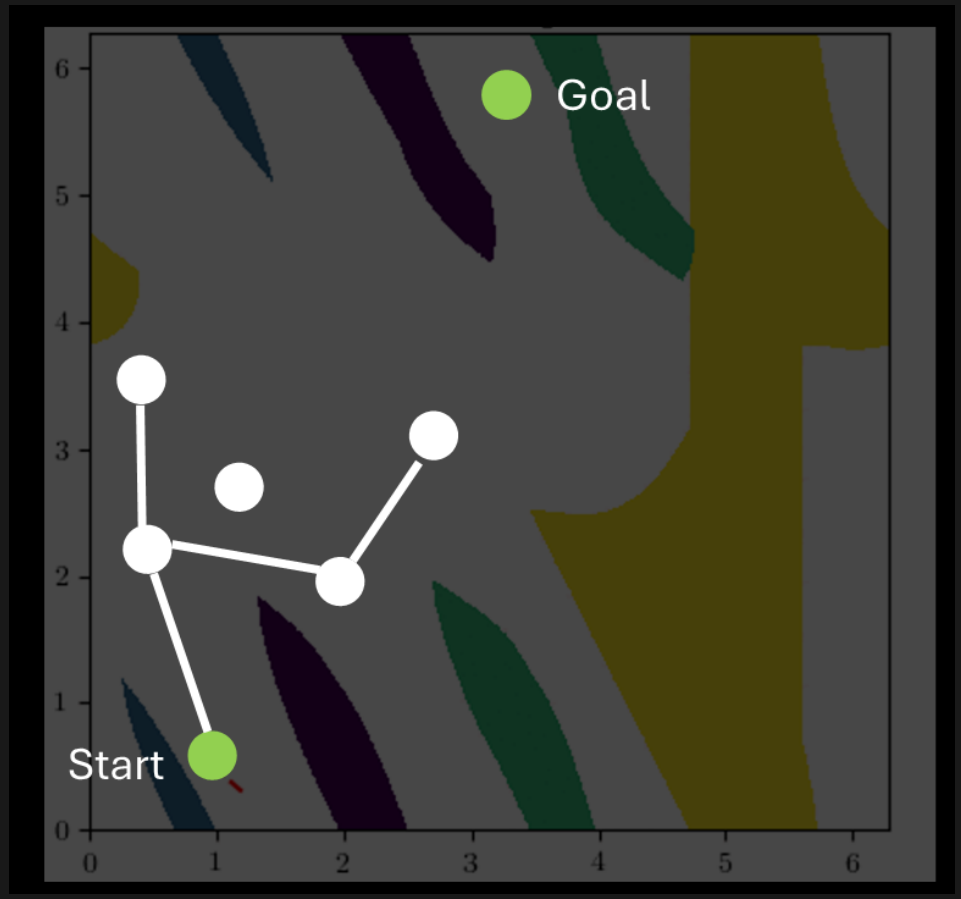
RRT Star (RRT*)



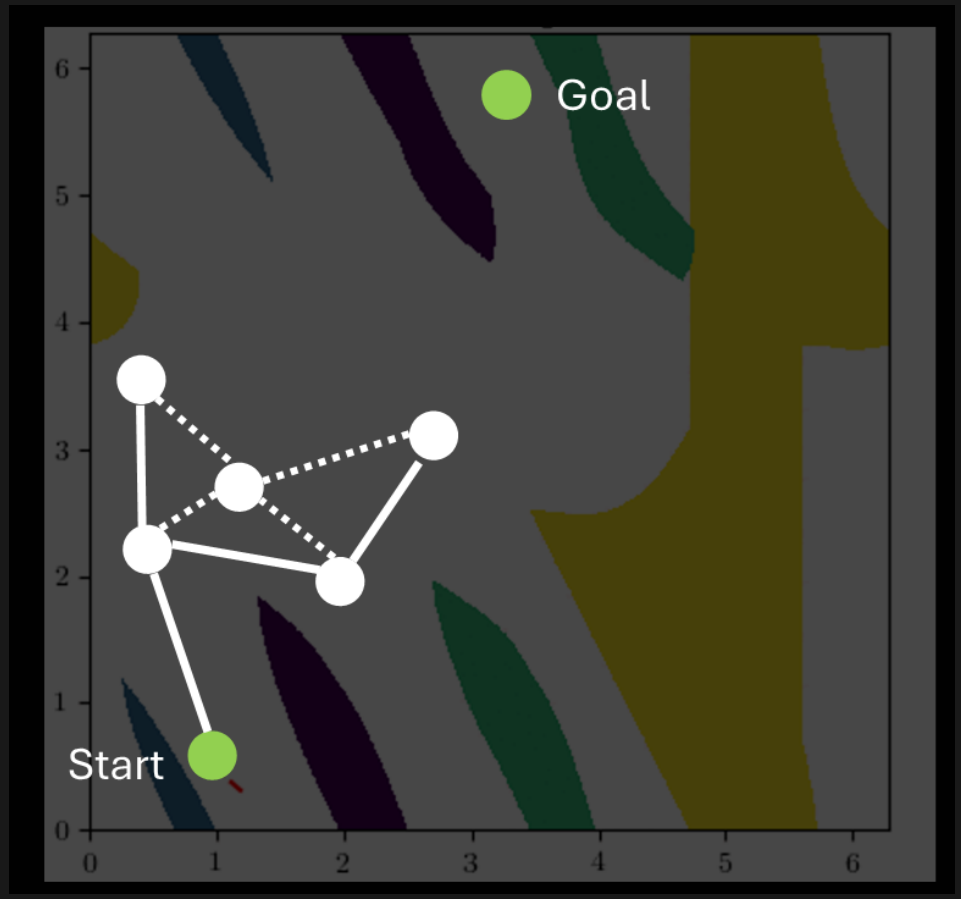
RRT Star (RRT*)



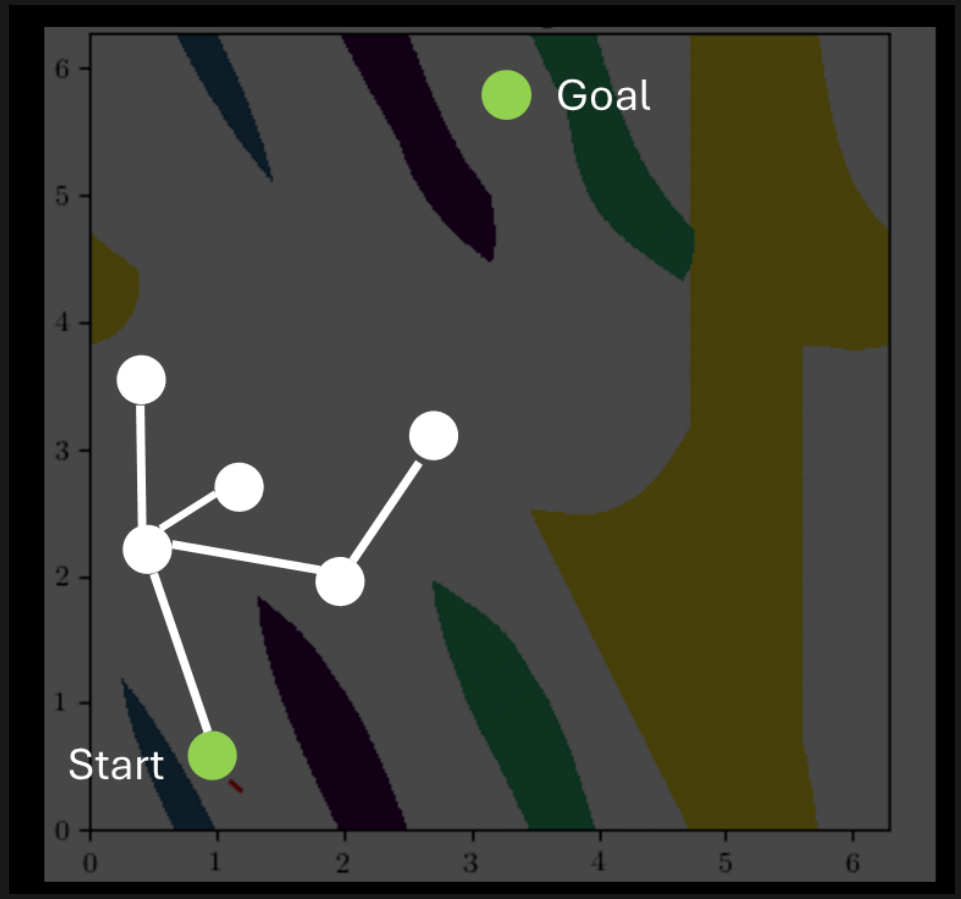
RRT Star (RRT*)



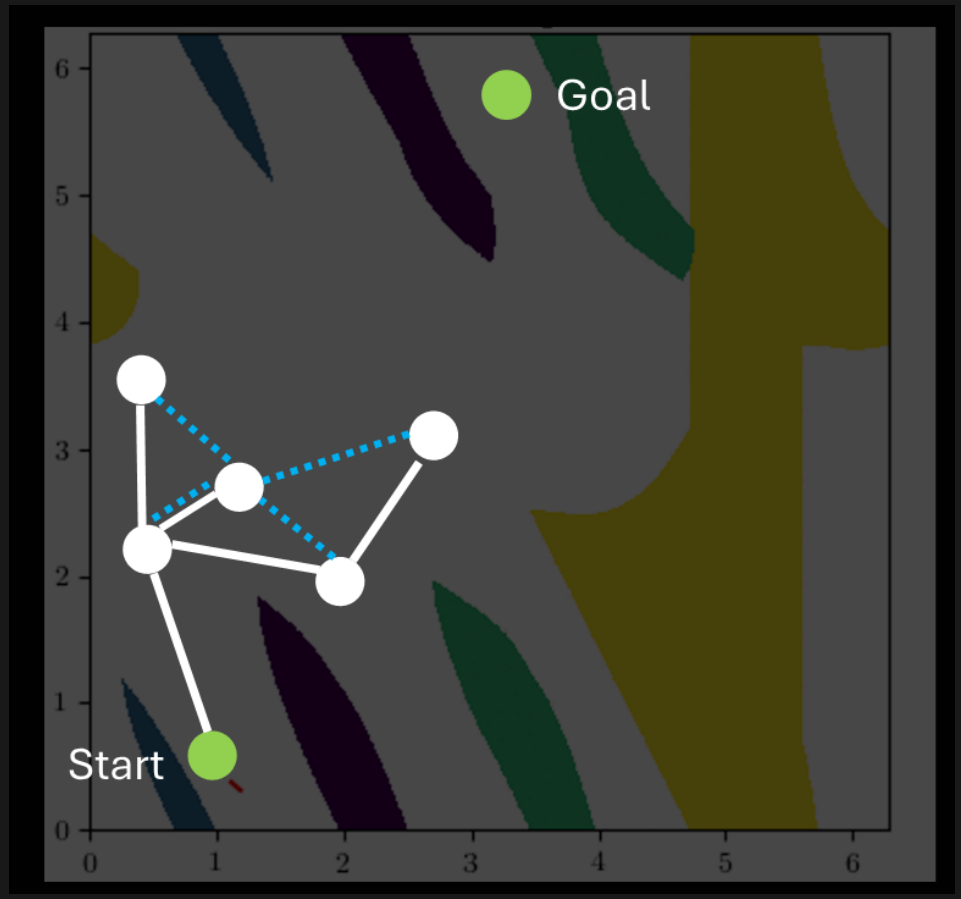
RRT Star (RRT*)



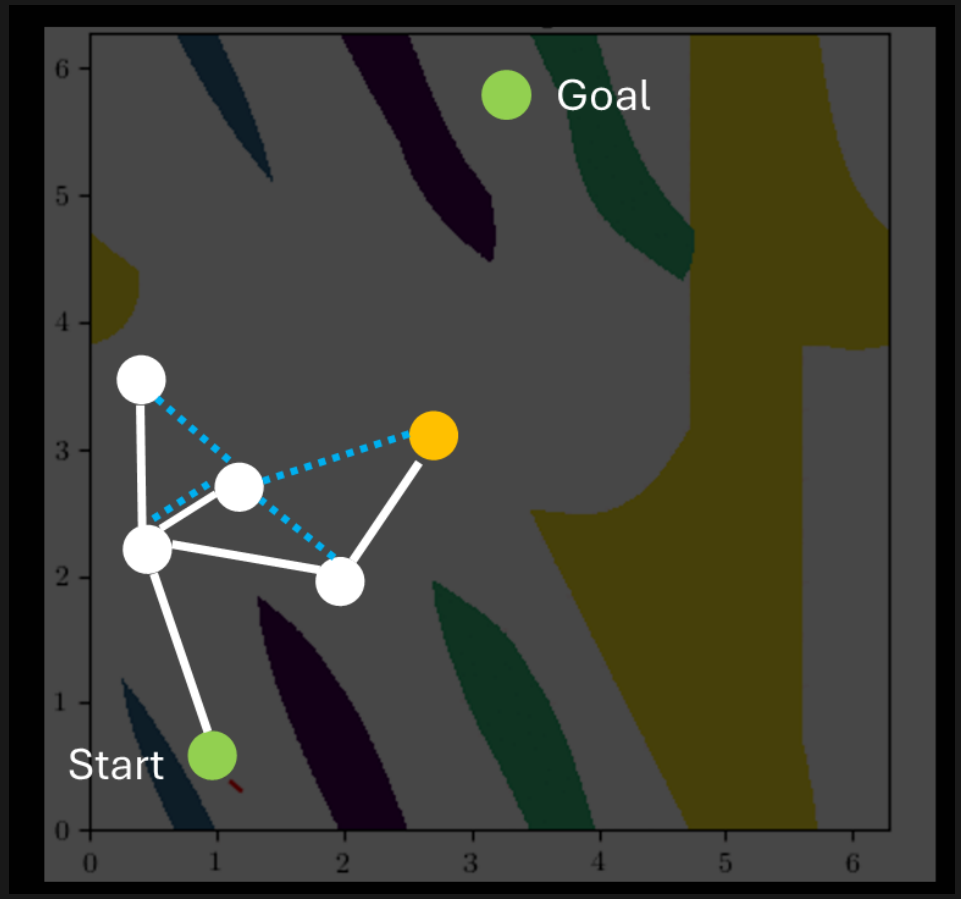
RRT Star (RRT*)



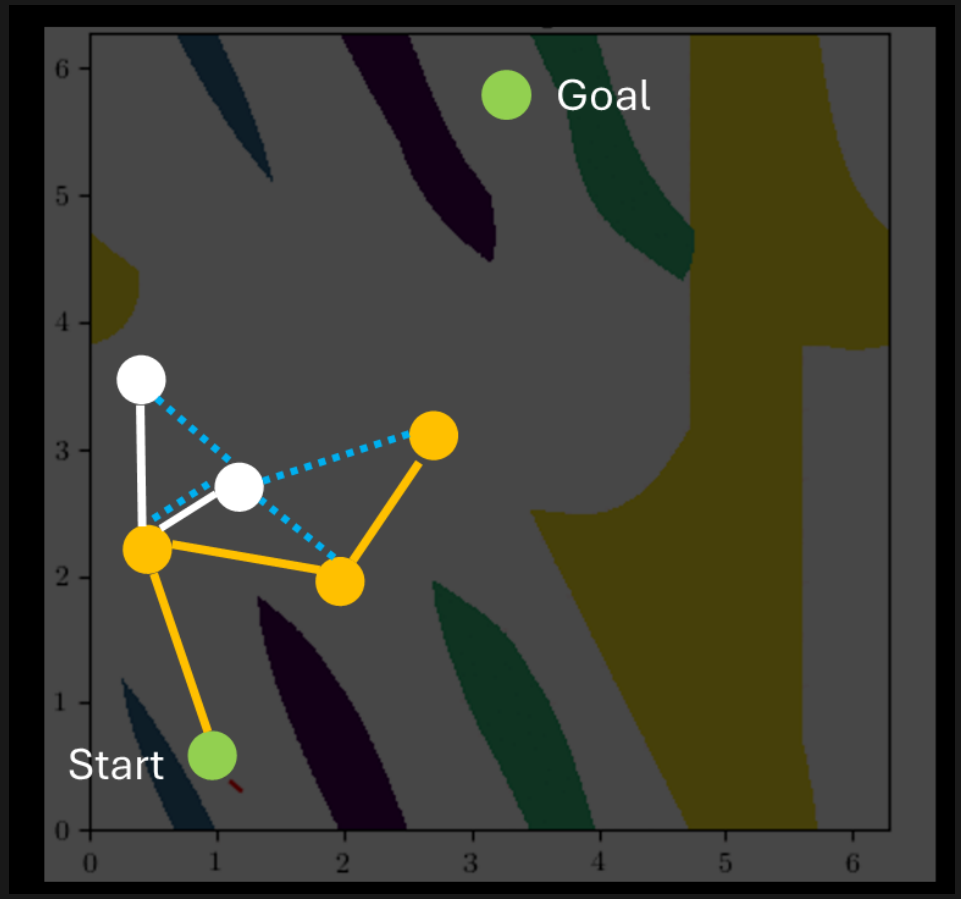
RRT Star (RRT*)



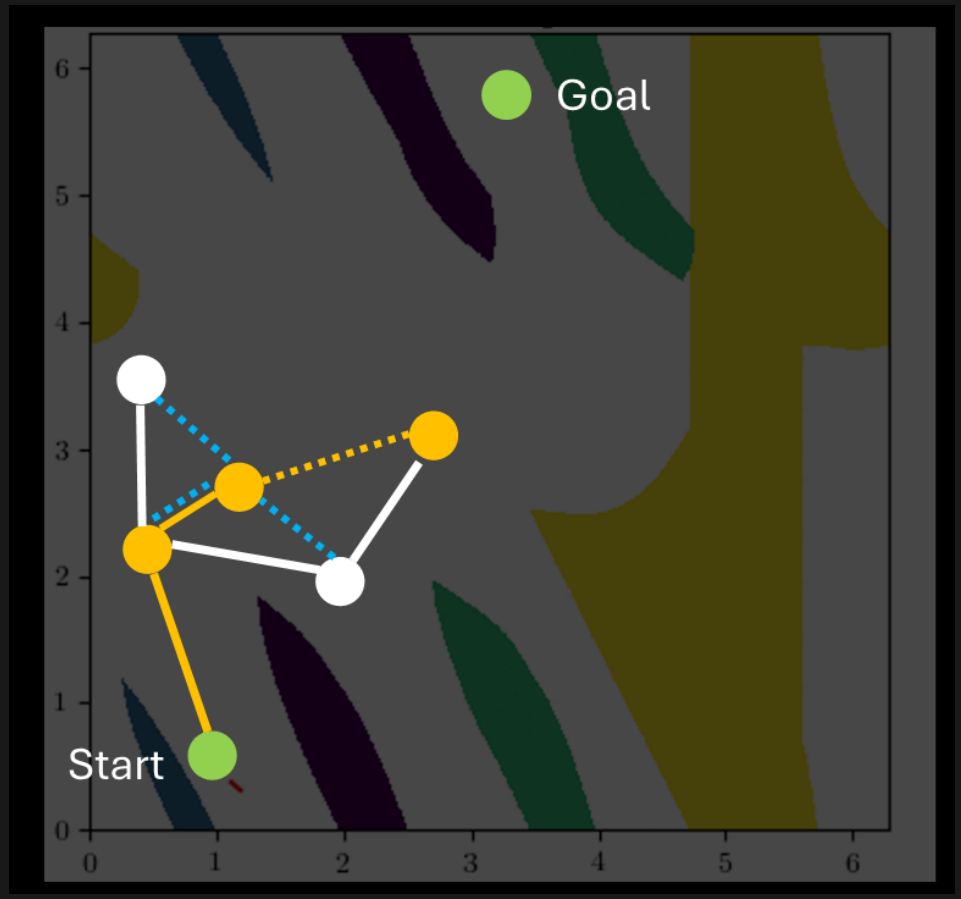
RRT Star (RRT*)



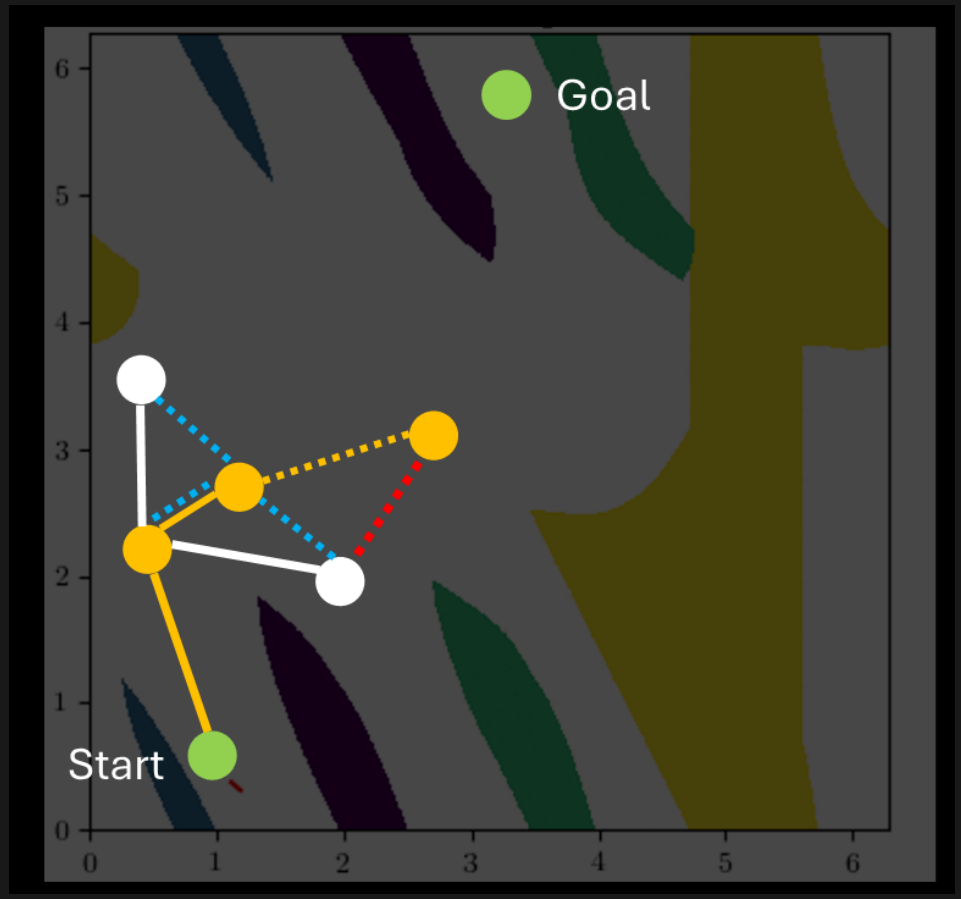
RRT Star (RRT*)



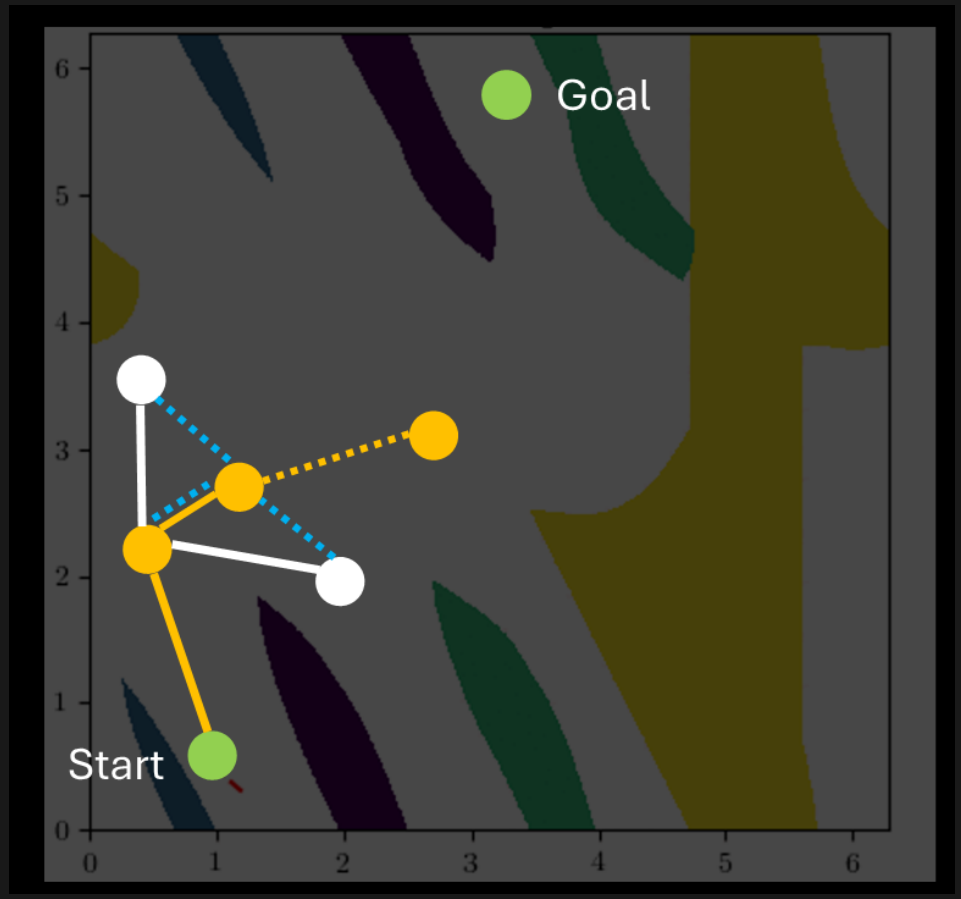
RRT Star (RRT*)



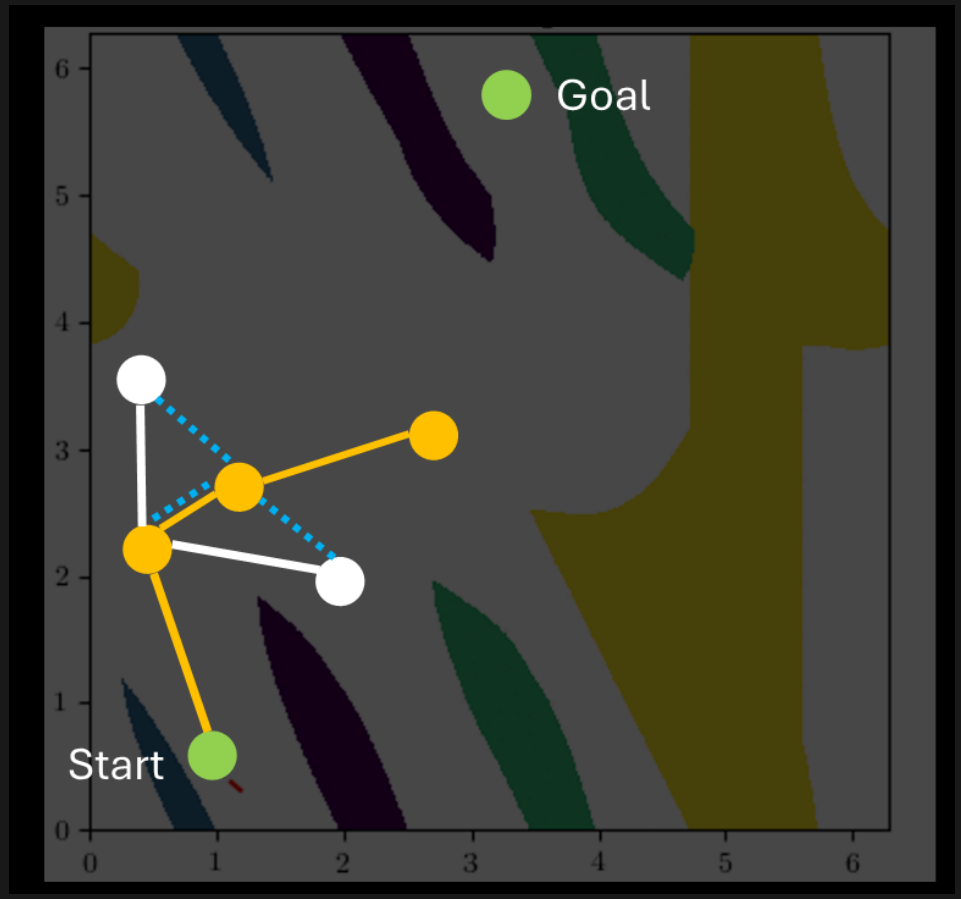
RRT Star (RRT*)



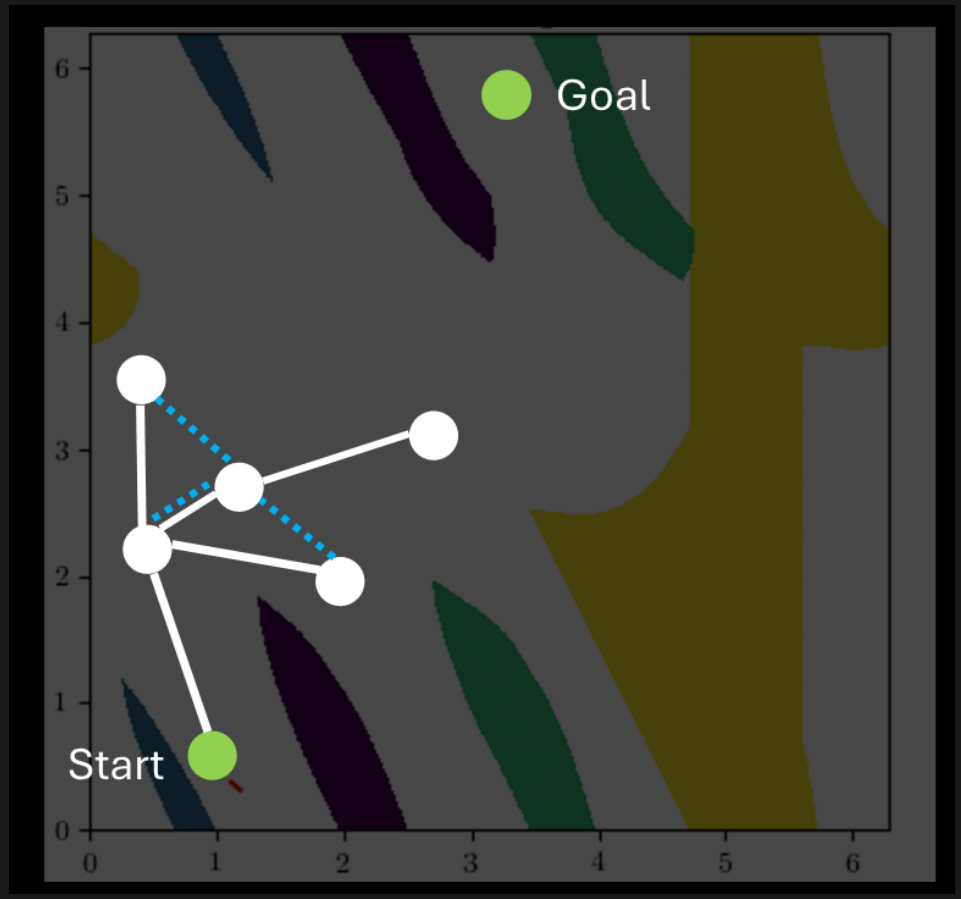
RRT Star (RRT*)



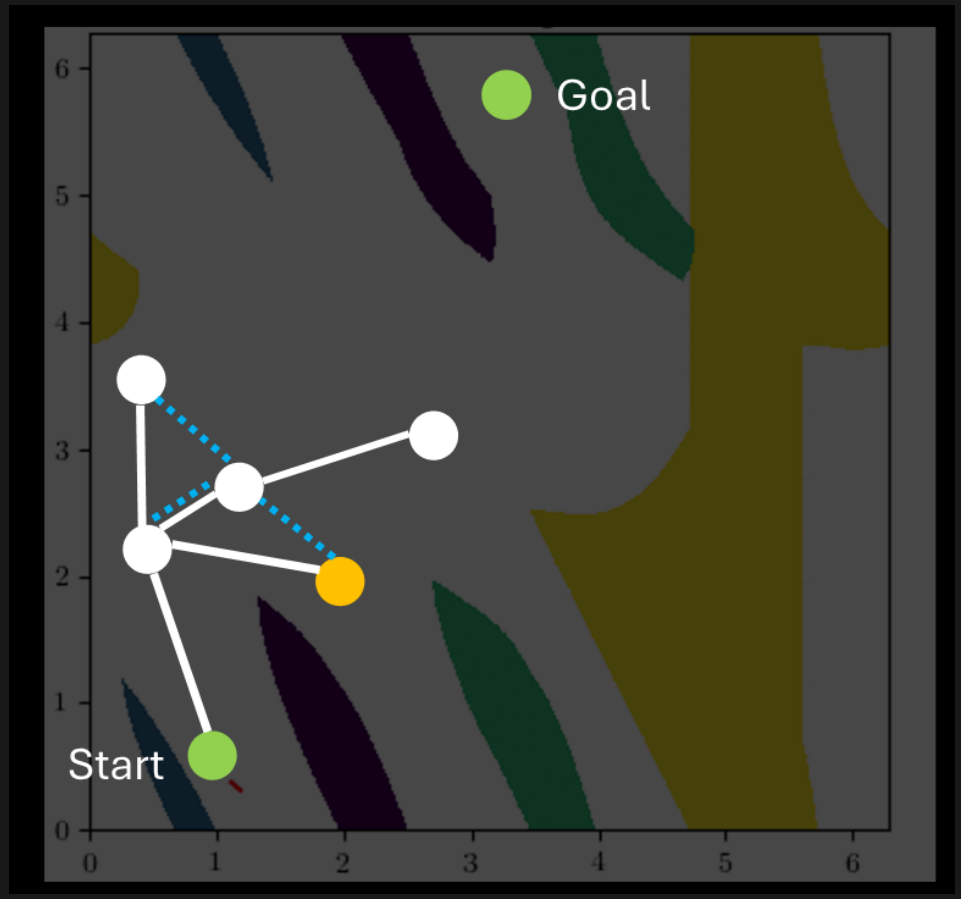
RRT Star (RRT*)



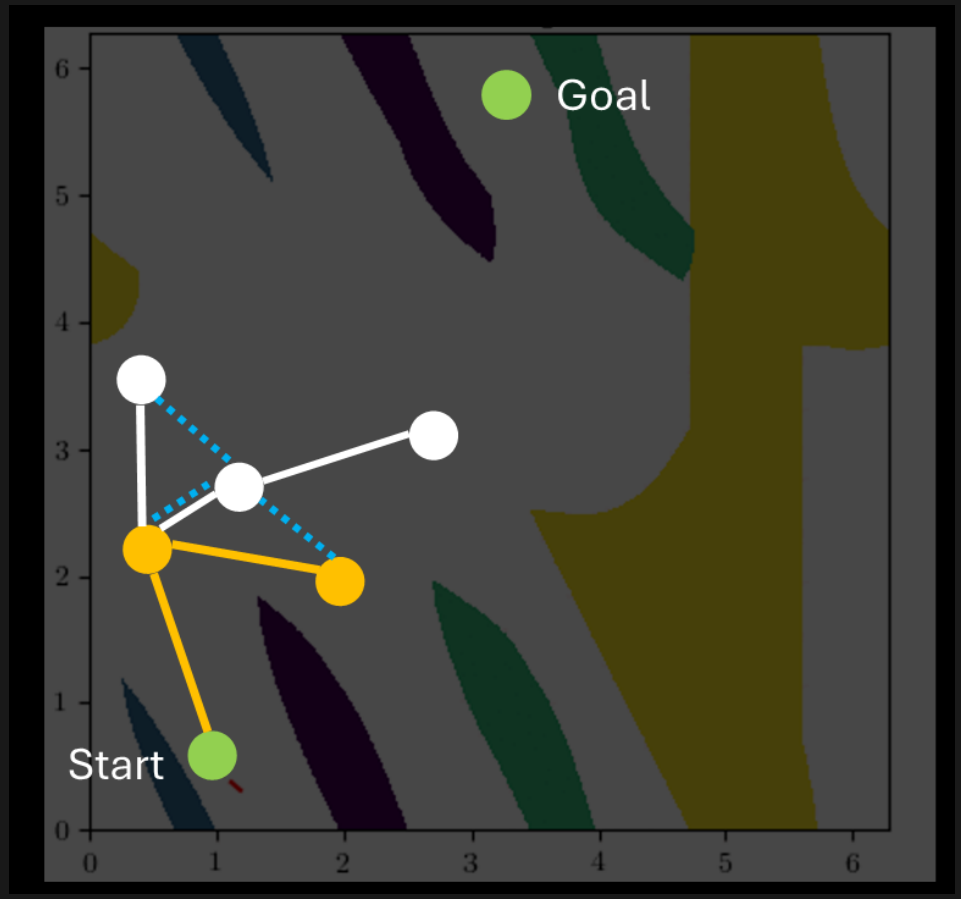
RRT Star (RRT*)



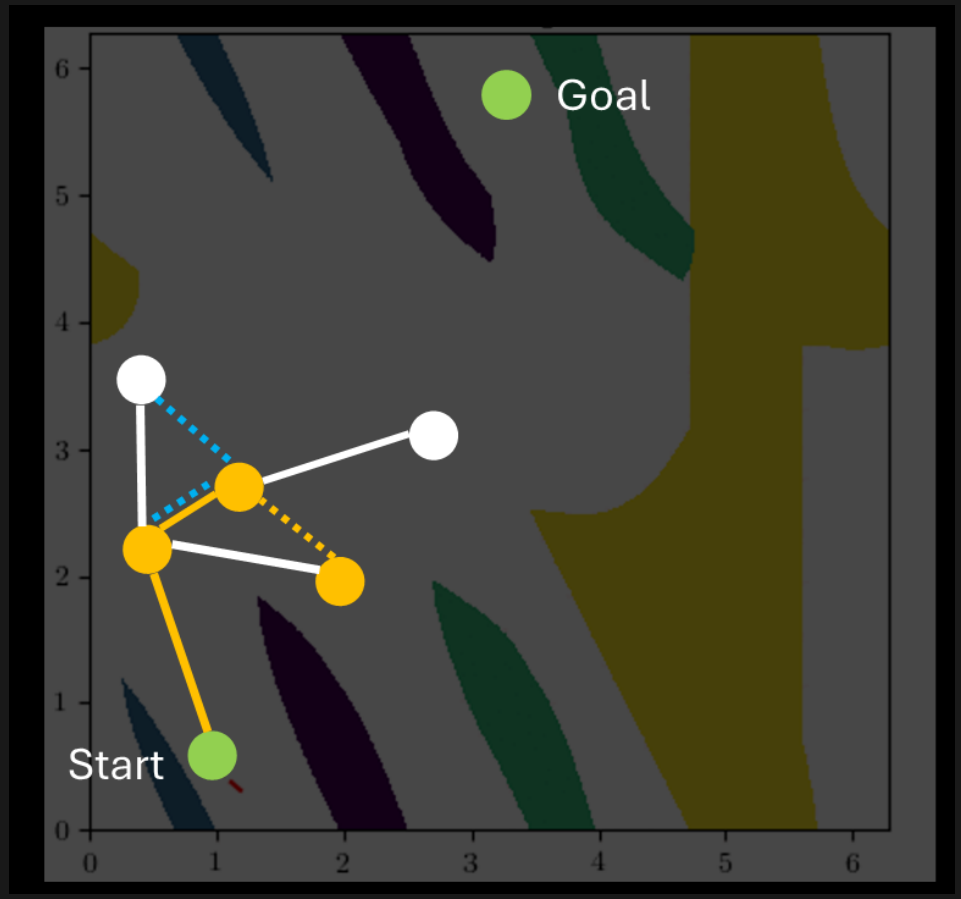
RRT Star (RRT*)



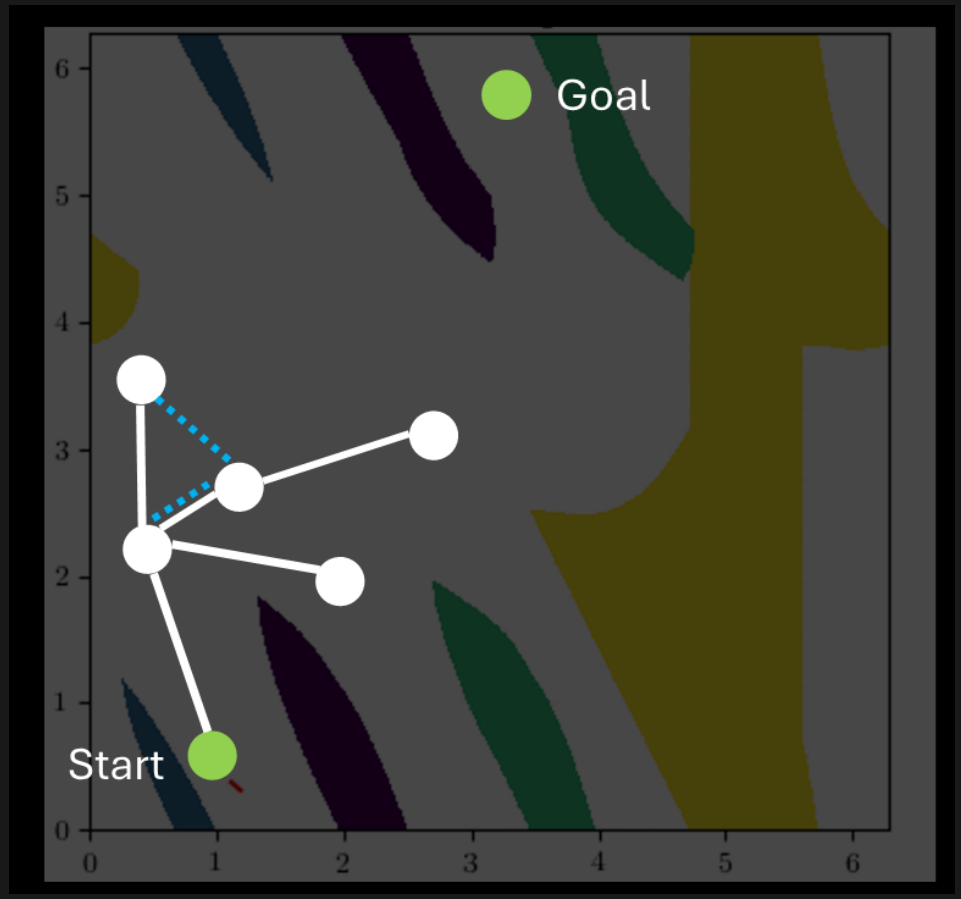
RRT Star (RRT*)



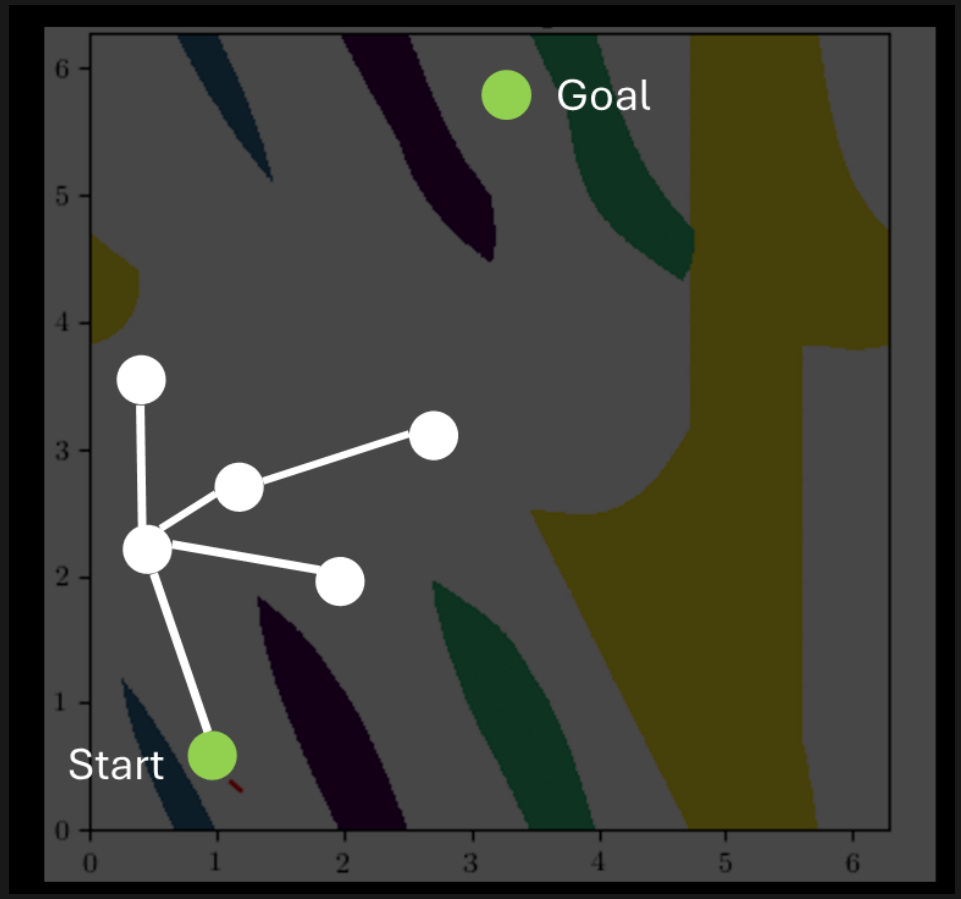
RRT Star (RRT*)



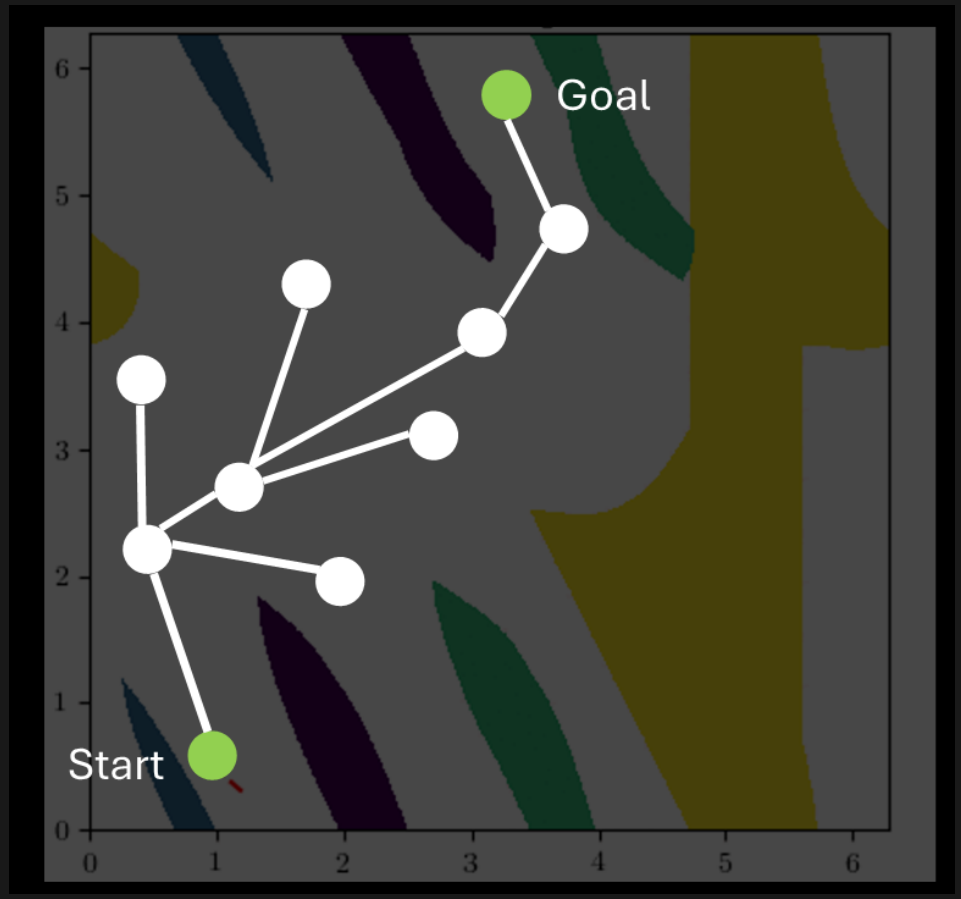
RRT Star (RRT*)



RRT Star (RRT*)



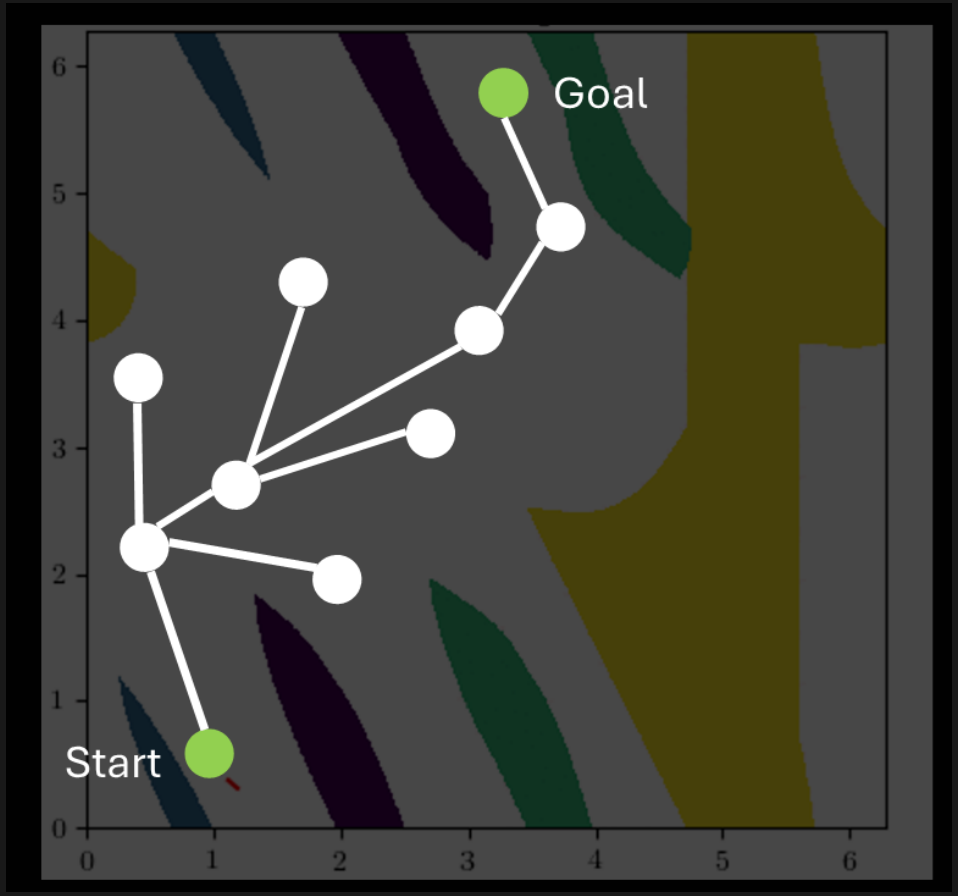
RRT Star (RRT*)



RRT Star (RRT*)

- 주요 특징
  - Single-query
  - Rewire

  - Narrow Passage
  - 최적의 경로 보장 X
  - Probabilistically Complete



RRT Star (RRT*)

Algorithm 4 Rapidly-exploring Random Tree (RRT)

Require: Maximum iterations N, step size δ, start q_{start}, goal q_{goal}
Ensure: A path from q_{start} to q_{goal}, if one exists

Initialize tree $T \leftarrow \{q_{\text{start}}\}$
for $i = 1$ to N **do**
  Sample random configuration q_{rand}
  $q_{\text{near}} \leftarrow \text{Nearest}(T, q_{\text{rand}})$
  $q_{\text{new}} \leftarrow \text{Steer}(q_{\text{near}}, q_{\text{rand}}, \delta)$
  if collision-free$(q_{\text{near}}, q_{\text{new}})$ **then**
    Add q_{new} to T with edge from q_{near}
    if $q_{\text{new}} \approx q_{\text{goal}}$ **then**
      return Extract path from q_{start} to q_{goal}
    end if
  end if
end for
return Failure (no path found)

Algorithm 9 Rapidly-exploring Random Tree Star (RRT*)

Require: Maximum iterations N, step size δ, start q_{start}, goal q_{goal}
Ensure: A near-optimal path from q_{start} to q_{goal}, if one exists

Initialize tree $T \leftarrow \{q_{\text{start}}\}$
$\text{cost}(q_{\text{start}}) \leftarrow 0$
for $i = 1$ to N **do**
  Sample random configuration q_{rand}
  $q_{\text{near}} \leftarrow \text{Nearest}(T, q_{\text{rand}})$
  $q_{\text{new}} \leftarrow \text{Steer}(q_{\text{near}}, q_{\text{rand}}, \delta)$
  if collision-free$(q_{\text{near}}, q_{\text{new}})$ **then**
    $Q_{\text{near}} \leftarrow \{\, q \in T : \|q - q_{\text{new}}\| \le r(|T|) \,\}$
    $q_{\min} \leftarrow \arg\min_{q \in Q_{\text{near}}} \big(\text{cost}(q) + \text{CostEdge}(q, q_{\text{new}})\big)$
    Add q_{new} to T with edge $(q_{\min}, q_{\text{new}})$
    $\text{cost}(q_{\text{new}}) \leftarrow \text{cost}(q_{\min}) + \text{CostEdge}(q_{\min}, q_{\text{new}})$
    for all $q \in Q_{\text{near}} \setminus \{q_{\min}\}$ **do**
      if $\text{cost}(q_{\text{new}}) + \text{CostEdge}(q_{\text{new}}, q) < \text{cost}(q)$ **then**
        Remove edge $\big(\text{Parent}(q), q\big)$
        Add edge (q_{new}, q)
        Update $\text{cost}(q)$
      end if
    end for
    if $q_{\text{new}} \approx q_{\text{goal}}$ **then**
      return Extract path from q_{start} to q_{goal}
    end if
  end if
end for
return Failure (no path found)

RRT vs. RRT*

RRT

- Tree
- Single-query

- Probabilistic Completeness
- Narrow Passage Problem
- Optimal 경로를 보장하지 않음

최단 경로 알고리즘

RRT*

- Tree
- Single-query
- “Rewire”

- Probabilistic Completeness
- Narrow Passage Problem
- Optimal 경로를 보장하지 않음
- RRT 보다는 더욱 짧은 경로를 보장

강의 요약

01

RRT*

- Tree
- Single-query
- Rewire

- Probabilistic Completeness
- Narrow Passage

02

알고리즘

03

코드 분석