FMT*

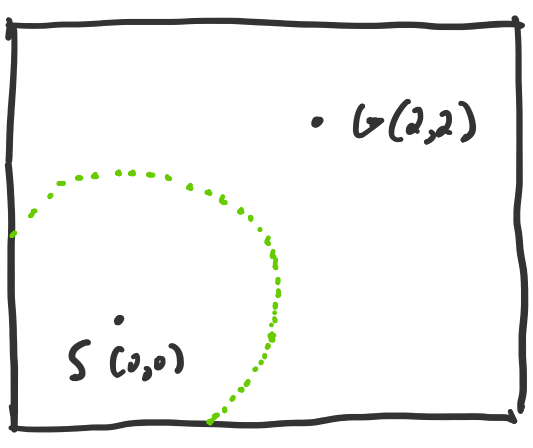
Tuesday, August 5, 2025 11:06 AM

Fast Marching Trees*

Core idea:

- FMT* randomly samples points in the space & builds a tree by expanding from the lowest-cost node
- As $N \to \infty$, converges to optimal path
- Only evaluates collision while connecting nodes (Lazy)

Simple Example

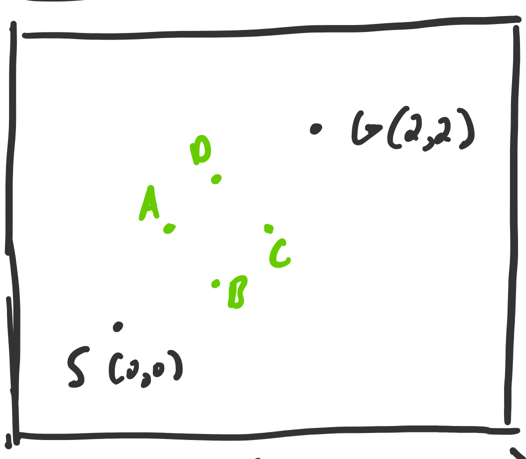


Start S: (0,0)

Goal G: (2,2)

Connection radius = 1

Step 1: Initialization



- Sample 5 points

A: (.5, 1)

B: (1, .5)

C: (1.5, 1.5)

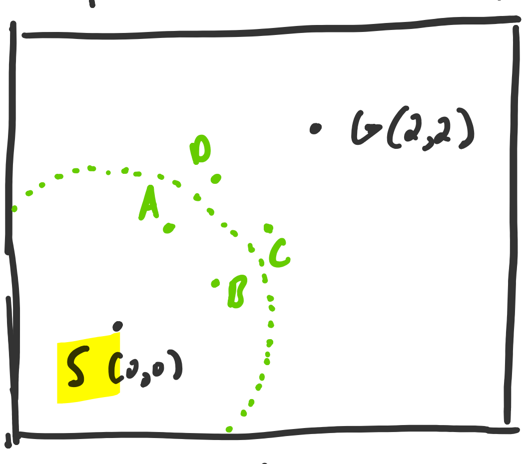
D: (1, 1.5)

Open = {S (cost=0)}

Unvisited = {A, B, C, D, G}

Step 2: Iteration 1

Explore lowest cost in open (S)



- Check cxns within 1.5 radius of S

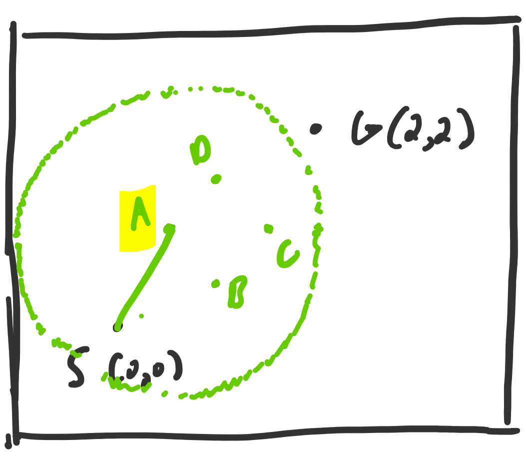
A: $d = \sqrt{(.5-0)^2 + (1-0)^2} = 1.12 < 1.5$ ✓

B: $d = \sqrt{(1-0)^2 + (.5-0)^2} = 1.12 < 1.5$ ✓

Open = {A(1.12), B(1.12)}

Unvisited = {C, D, G}

Step 3: Iteration 2 – Explore A



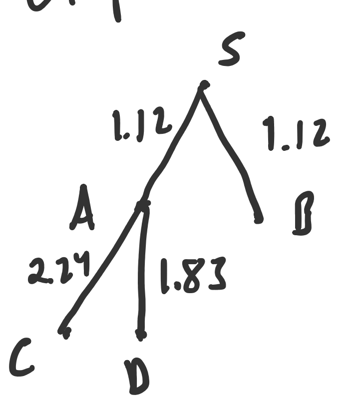
- Check cxns < 1.5 of A

C: $d = 1.12 \leq 1.5$ ✓

D: $d = .71 \leq 1.5$ ✓

G: $d = 1.8 > 1.5$ ✗

Graph Representation



Open = {B(1.12), C(2.24), D(1.83)}

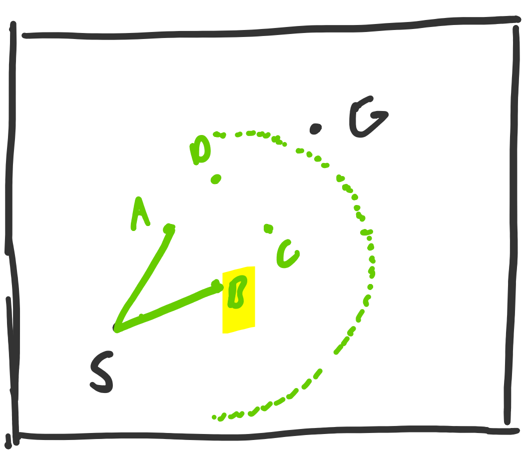
From S so same val — 1.12+1.12 — 1.12+.71

Unvisited = {G}

Step 4: Iteration 3 Expand B (1.12)

✓ Lowest cost

Check distances from B (1, .5) to all unvisited nodes (G)



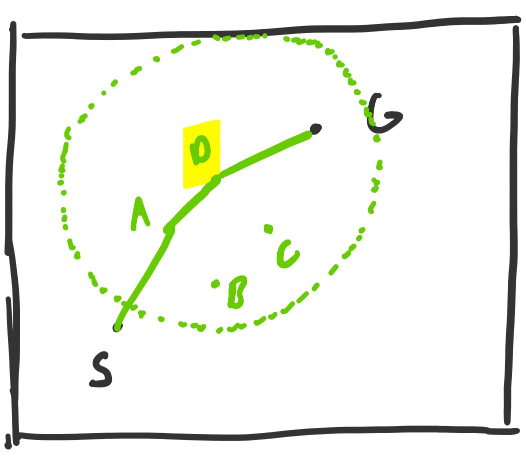
G(2,2): $d = \sqrt{(2-1)^2 + (2-.5)^2} = 1.8 > 1.5$ ✗

So, B was not fruitful... skip!

Open = {D(1.83), C(2.24)}

Unvisited = {G}

Step 5 – Iteration 4 Expand D (1.83)



G(2,2): $\sqrt{(2-1)^2 + (2-1.5)^2} = 1.12 < 1.5$ ✓

Goal Reached!

Update cost: $D_{cost} + \overrightarrow{DG} = 1.83 + 1.12 = 2.95$

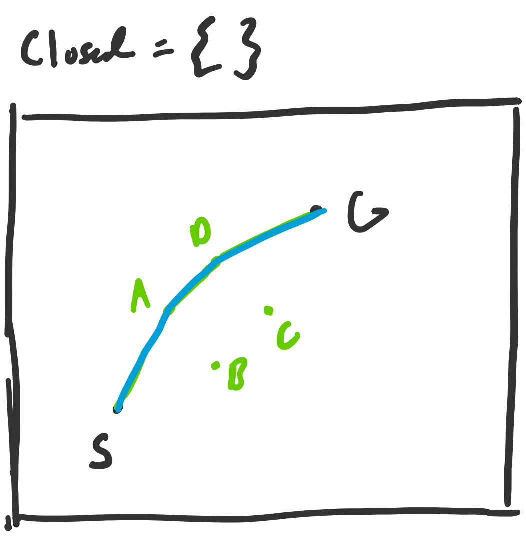
Open = {C(2.24), G(2.95)}

Closed = {}

Step 6

Open = {C(2.24), G(2.95)}

Closed = {}



b/c unvisited list is empty, algorithm terminates

Path: S → A → D → G