The Art Gallery Theorem

Simple Polygon

- Let $v_0, v_1, v_2, \ldots, v_{n-1}$ be n points in the plane
- Let $e_0 = v_0v_1, e_1 = v_1v_2, \ldots, e_{n-1} = v_{n-1}v_0$ be n segments connecting the points.

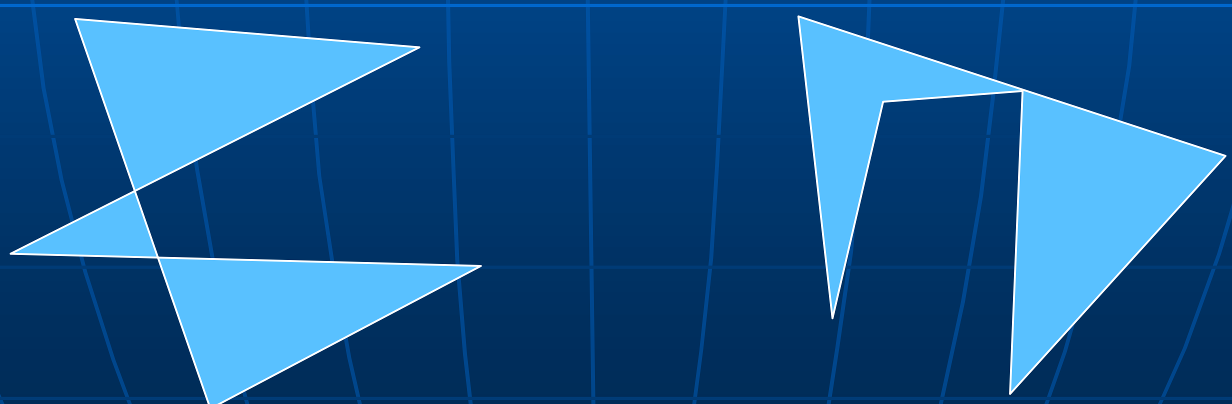
Definition

1. Adjacent segments share a single common point
   - $e_i \cap e_{i+1} = v_{i+1}$, for all $i = 0, \ldots, n-1$
2. Nonadjacent segments do not intersect
   - $e_i \cap e_j = \Phi$, for all $j \neq i+1$

Example

- Simple Polygon

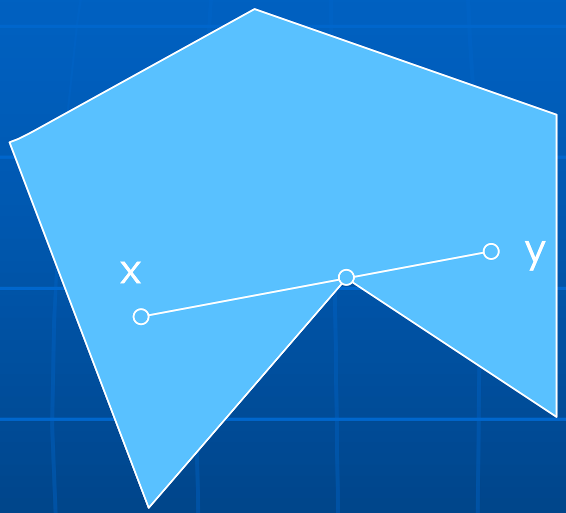
- Nonsimple Polygon



Art Gallery Theorem

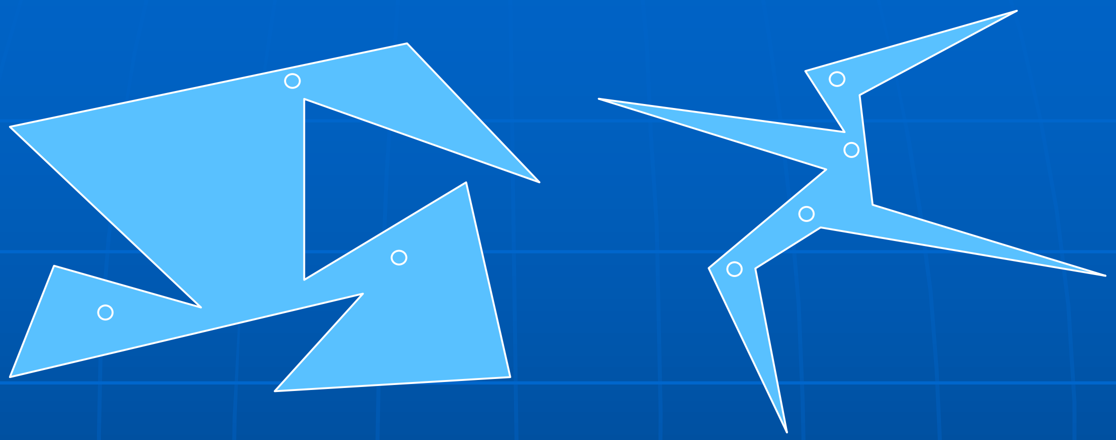
- n개의 vertices로 이루어진 Art Gallery가 있음
  - 모든 vertices를 감시할 수 있는 경비원을 배치하여야 할 때 필요한 최소의 경비원 수는?
  - Art Gallery의 모든 부분을 밝힐 수 있는 최소의 조명 수는?

Art Gallery Theorem



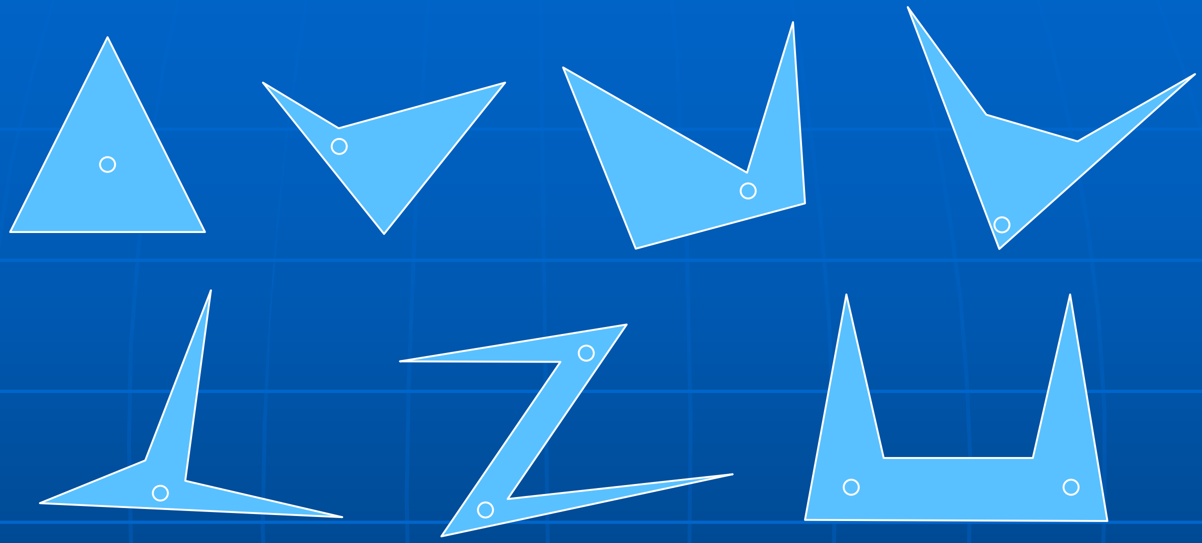
- X에서 y를 볼 수 있다고 가정
- 경비원은 투명하다고 가정

Art Gallery Theorem



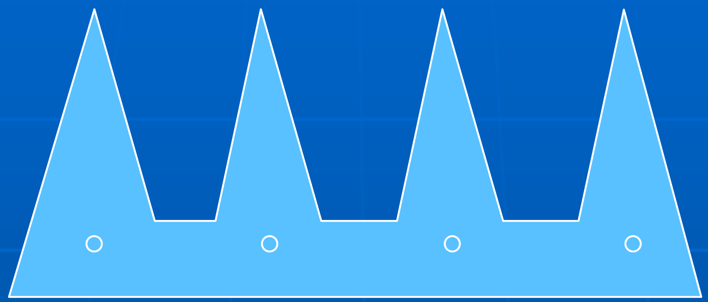
- 12개의 vertices를 가진 다른 모양의 두 Art Gallery
- 왼쪽은 세 명의 경비원, 오른쪽은 네 명의 경비원이 필요
- Max over min formulation
  - n개의 vertices로 이루어진 여러 모양의 polygon에서 각각 최소의 필요한 경비원 수 중의 최대값을 취함
  - 그 최대값을 G(n)이라고 하자

Art Gallery Theorem



- $1 \leq G(n) \leq n$
- $G(3)=1$
- $G(4)=1$
- $G(5)=1$
- $G(6)=2$

Lower bound of ⌊n/3⌋



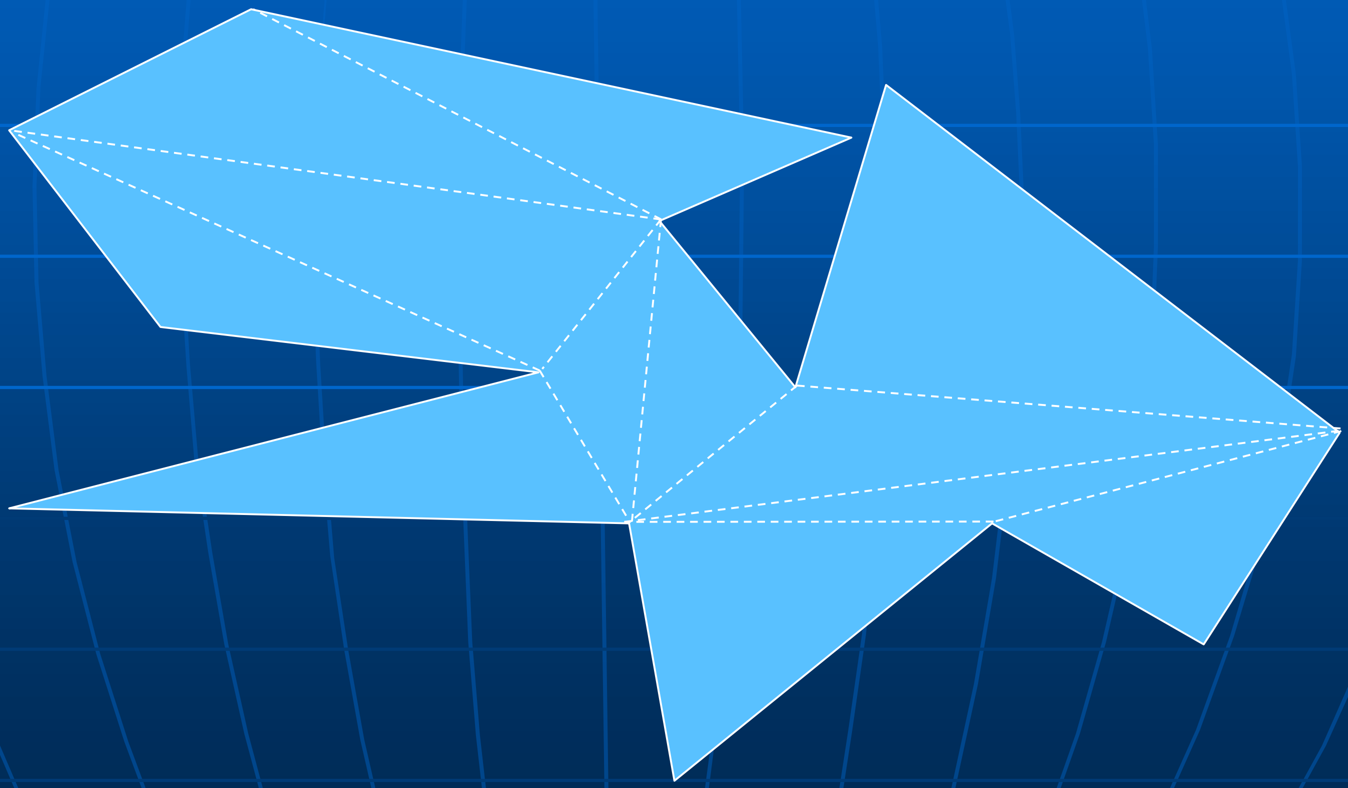
- k개의 prongs(뾰족한 끝)을 가진 comb(빗)모양의 polygon을 생각해 보자
  - 각 prong마다 한명의 경비원을 세우면 된다
  - k개의 prongs이 있는 comb은 3k개의 vertices를 가진다.
  - G(n) >= $\lfloor n/3 \rfloor$

Fisk's Proof of Upper-bound: Three Coloring

- n개의 vertices로 이루어진 임의의 polygon P가 있다고 가정

Three Coloring

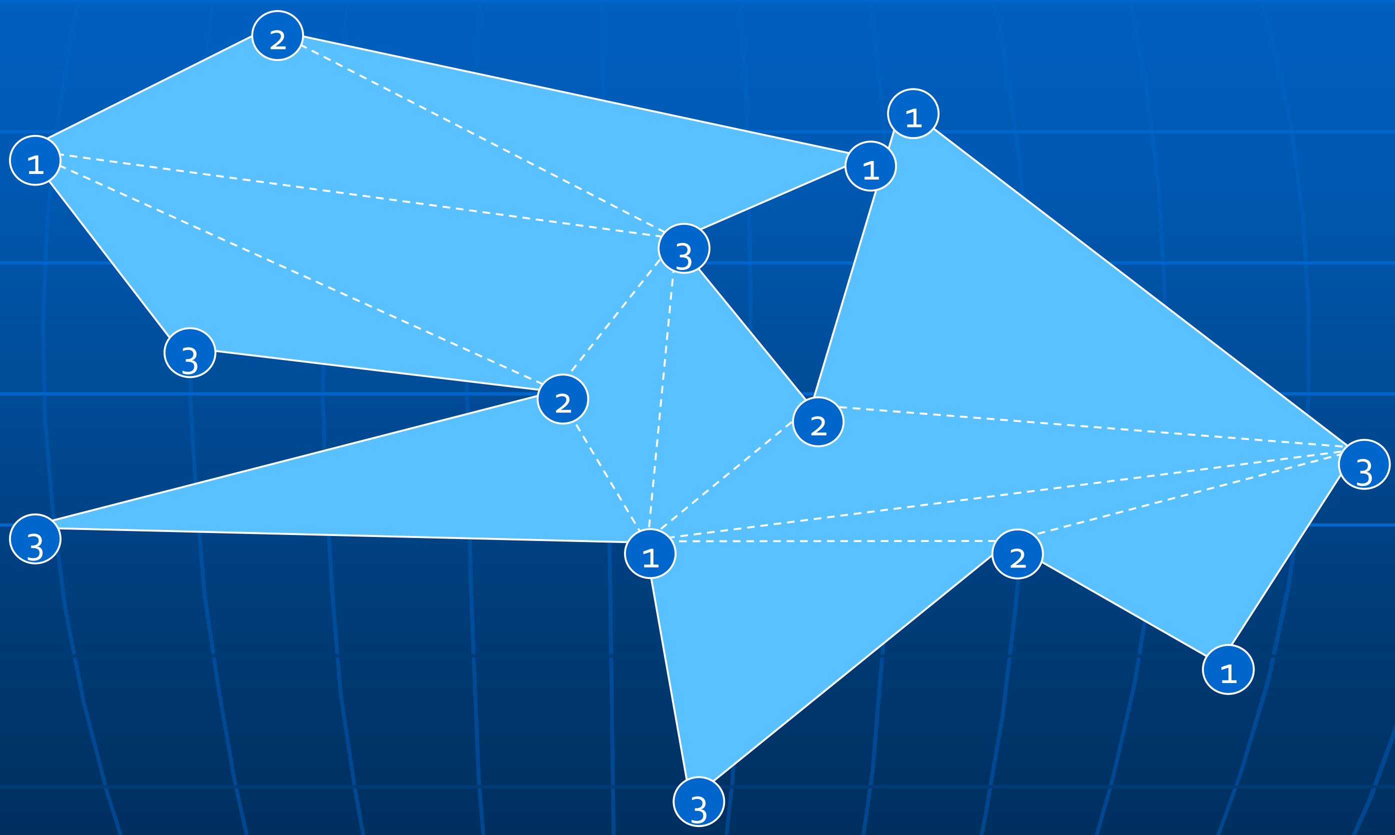
- Step 1. Triangulate P



Three Coloring

- Step2. 3-coloring
  - edge와 diagonal은 arc
  - vertices는 node
  - 모든 triangulation graph는 3-colorable

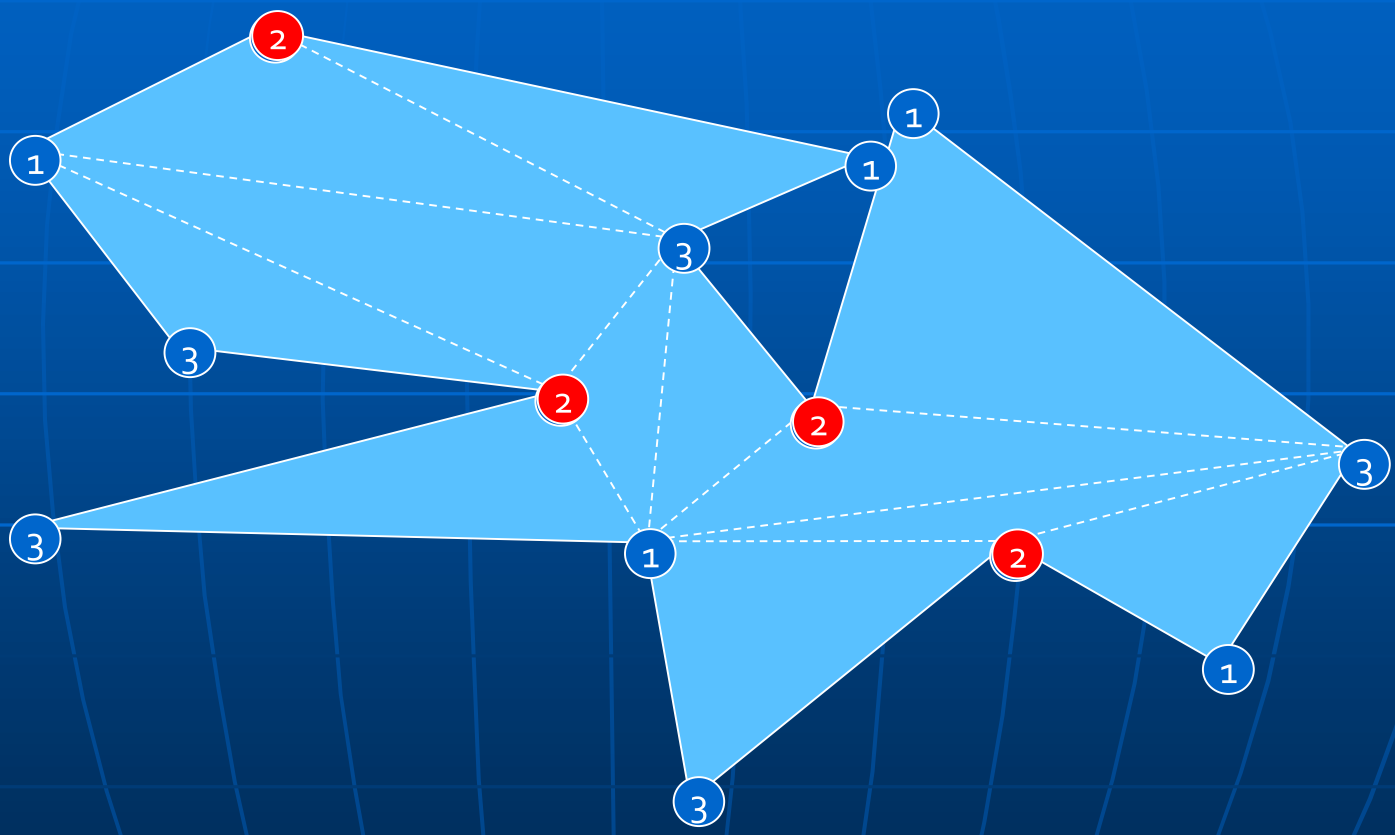
Three Coloring



Three Coloring

- Step 3. 가장 적게 사용된 색의 vertex에 경비원을 세움
  - 삼각형은 경비원을 한 vertex에만 세우면 커버됨
  - 모든 삼각형 vertex는 3개의 서로 다른 색으로 칠해져 있음
  - 어느 한 색에 경비원을 세운다면 모든 삼각형 커버할 수 있다는 게 보장됨
  - 따라서 가장 적게 사용된 색의 vertex에 경비원을 세우는 게 optimal!

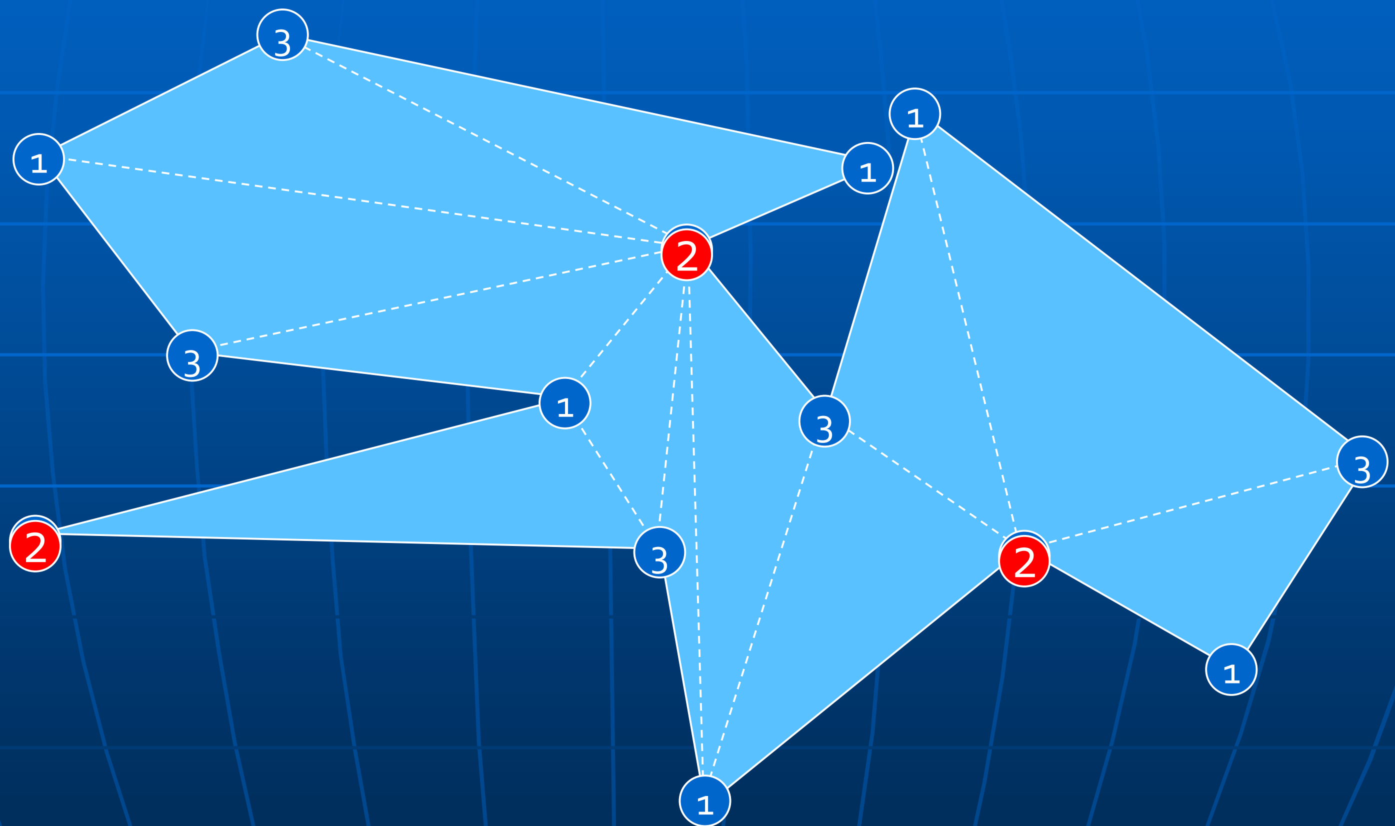
Three Coloring



Three Coloring

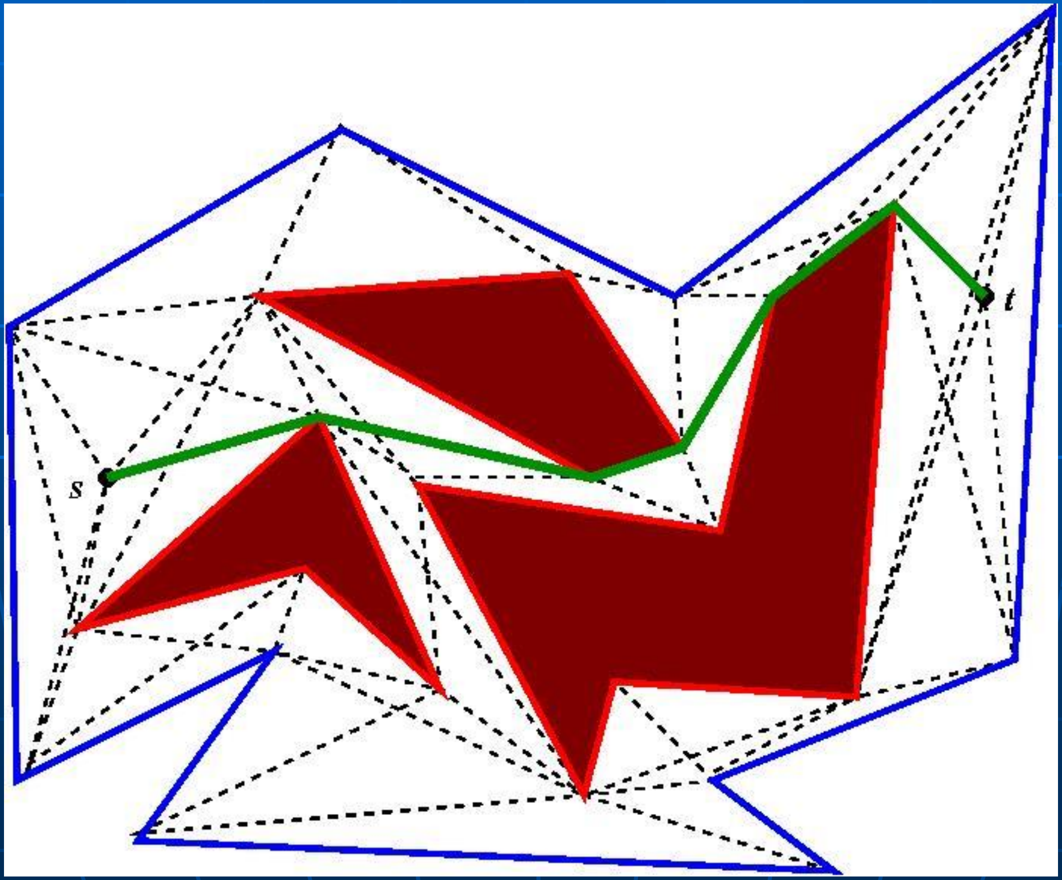
- Step 4. 비둘기 집 원리를 적용
  - n개의 vertices가 3개의 색에 모두 들어가야 함
  - 가장 적게 사용되는 색은 $\lfloor n/3 \rfloor$ 이하 사용됨
  - $G(n) \leq \lfloor n/3 \rfloor$
  - 단, G(n)은 꼭지점 n개를 가진 모든 다각형에 대한 최대값 => 각 다각형의 최소 경비원수는 더 적어질 수도 있다!!

Three Coloring

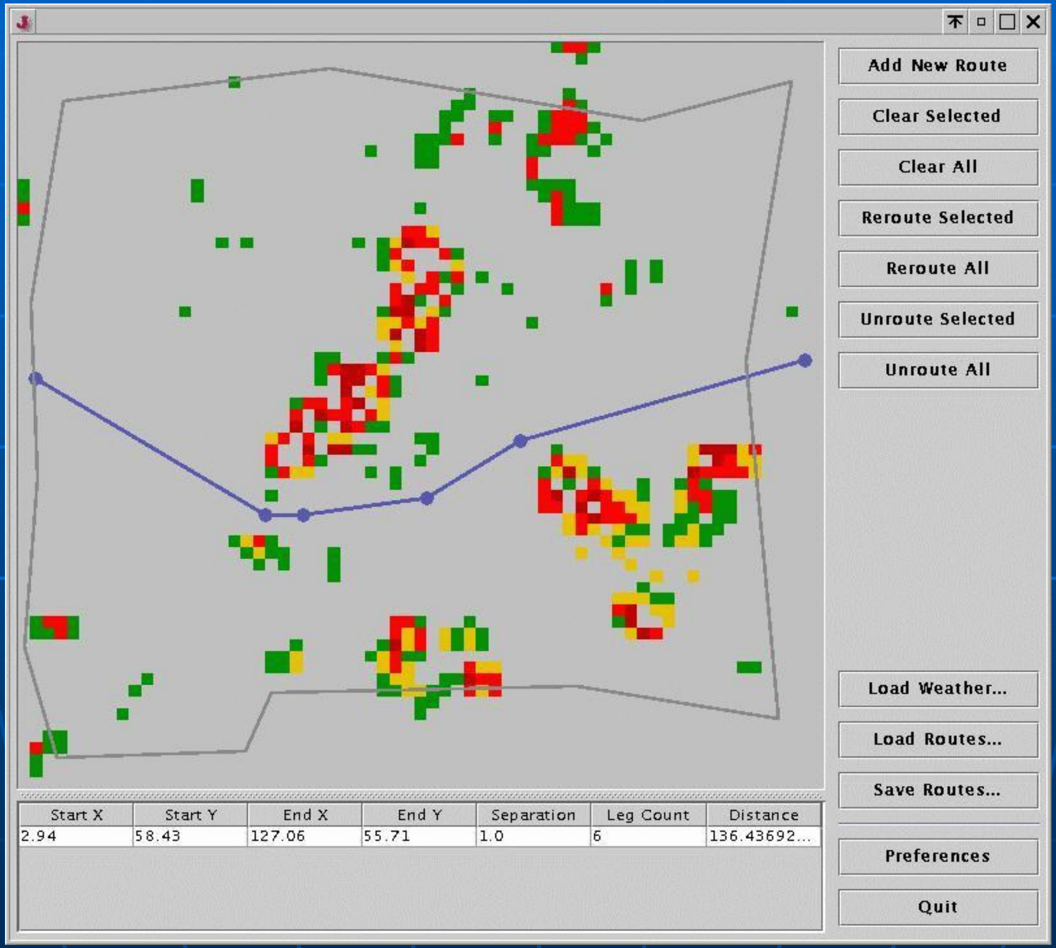


Robot Motion Planning

Example: Shortest path for a mobile robot

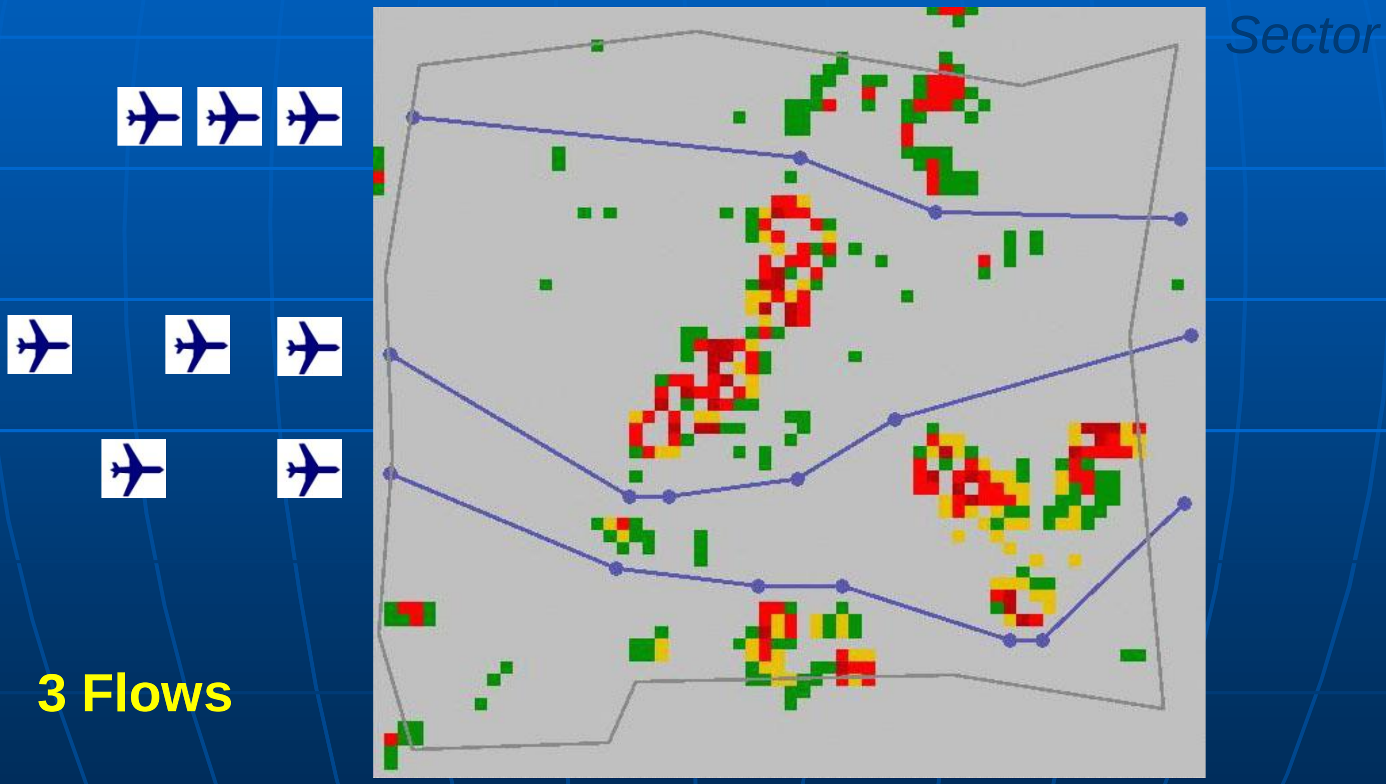


Application: Weather Avoidance



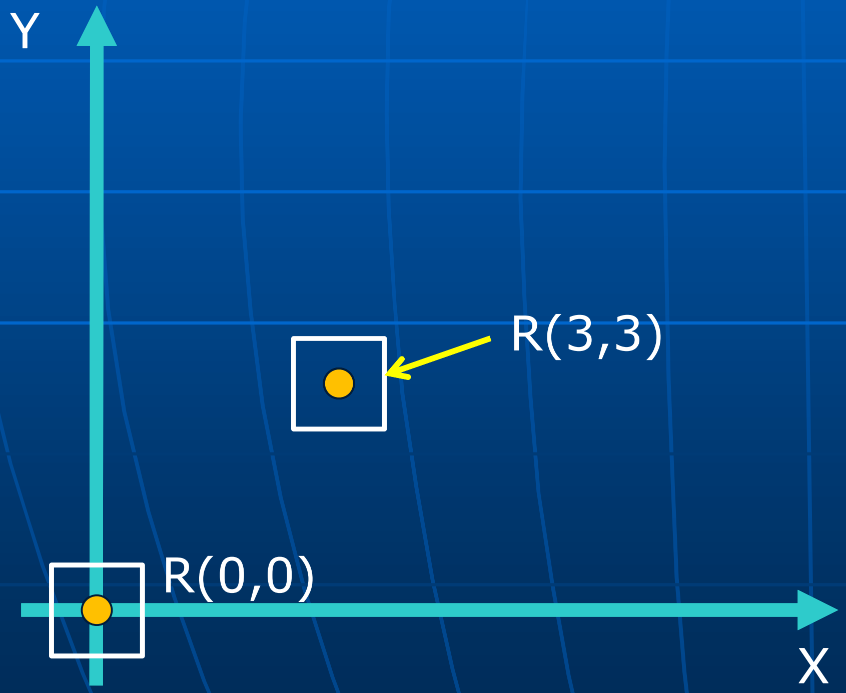
From Lecture notes on CG by Joe Mitchell at Stony Brook

Weather Avoidance Algorithms for En Route Aircraft



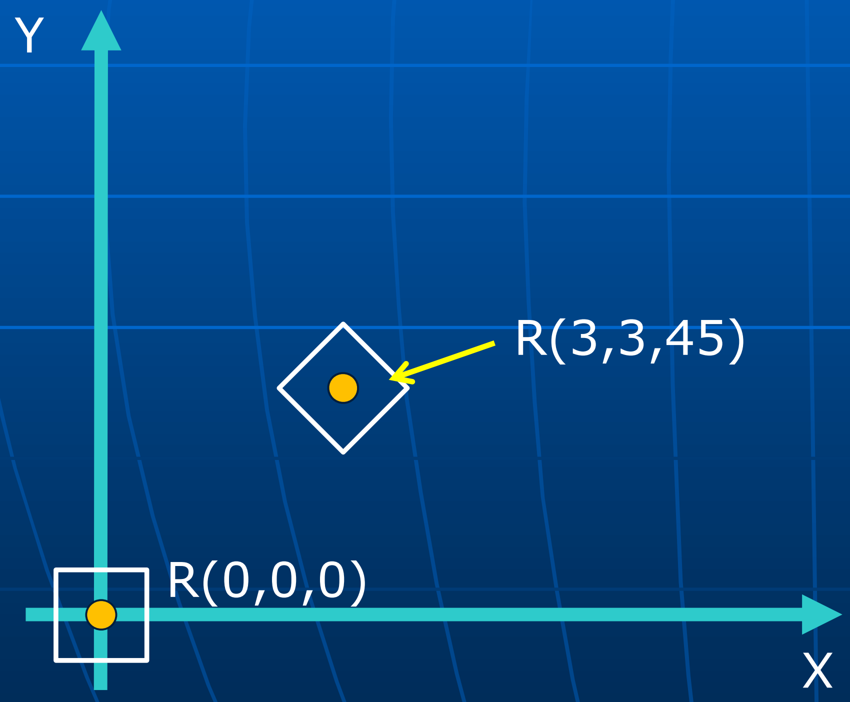
Robot Motion Model vs. Configuration space

2-d Translation-only robot

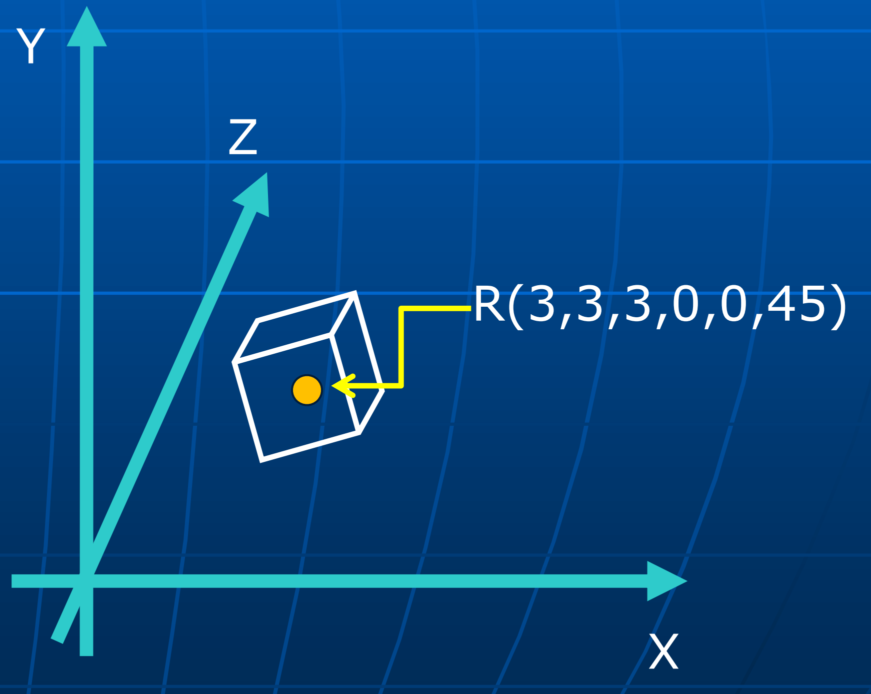


Robot Motion Model vs. Configuration space

2-d Rotation-allowed robot

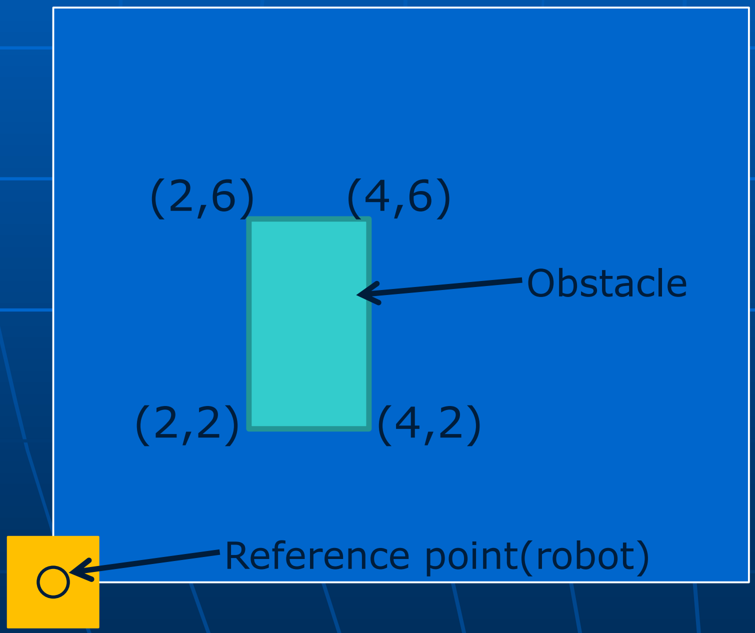


3-d Rotation-allowed

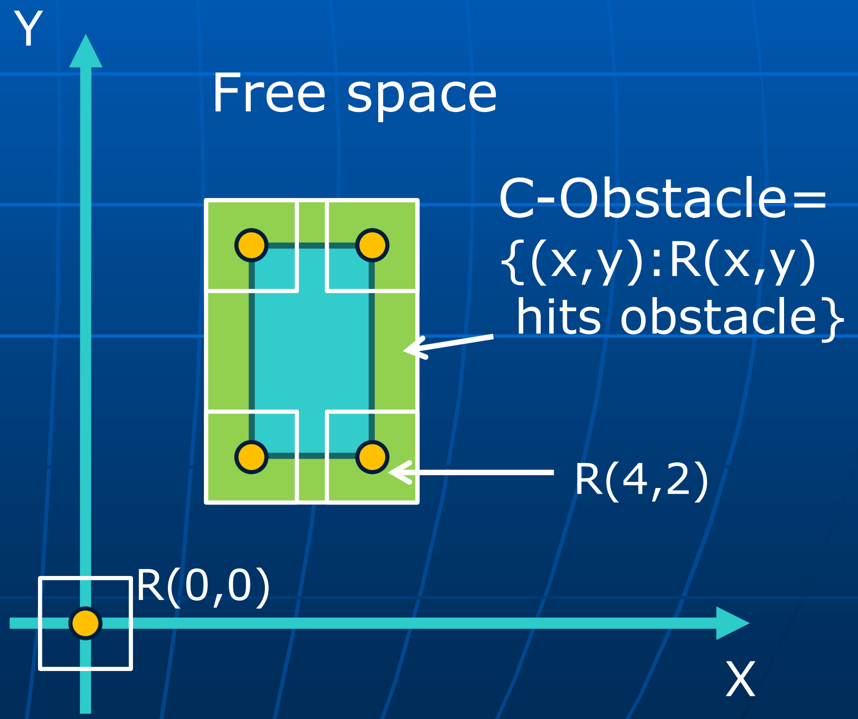


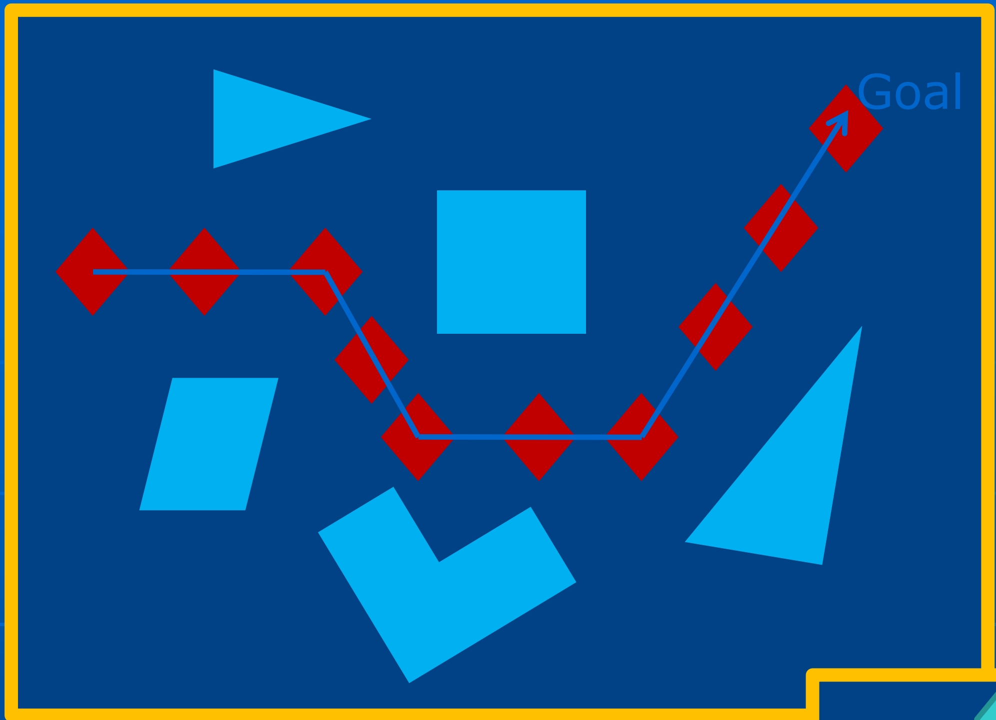
Translation-Only Robot Motion

Work space
(Real world)



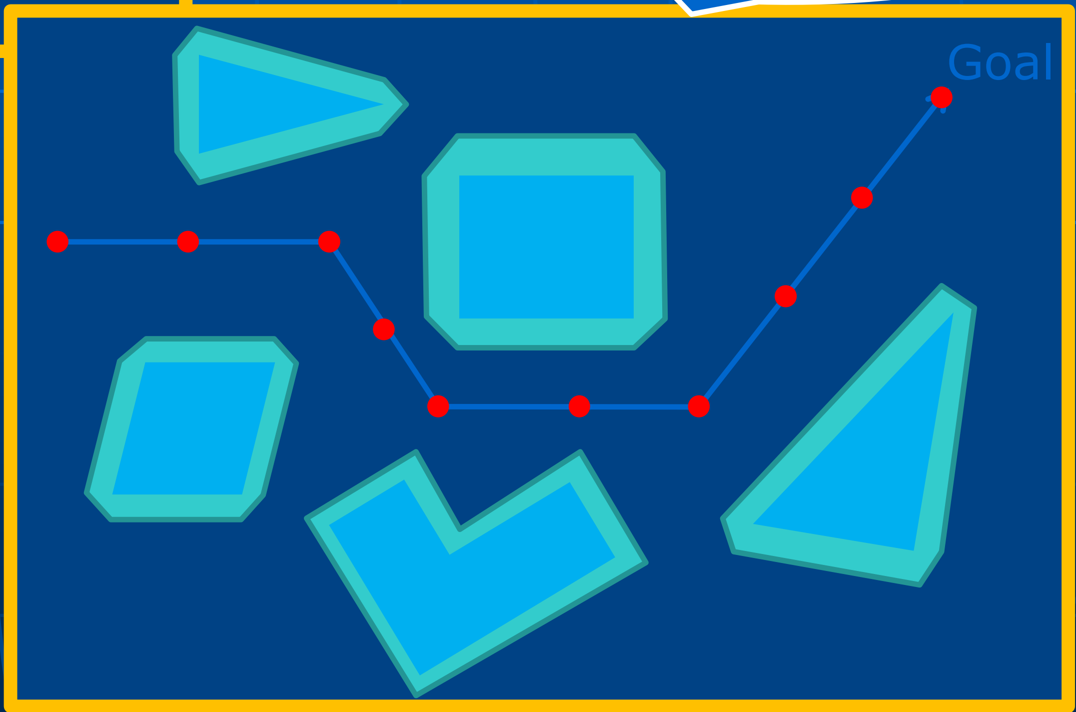
Configuration space
(parameter space)





Path in Work Space

Free Space



Path in Configuration Space

Translation-Only Robot

- Minkowski sum of two sets A and B

  $$A \oplus B = \{a + b : a \in A, b \in B\}$$

- Theorem

- P: obstacle, R: planar translating robot

  C-obstacle of P is $P \oplus (-R(0,0))$

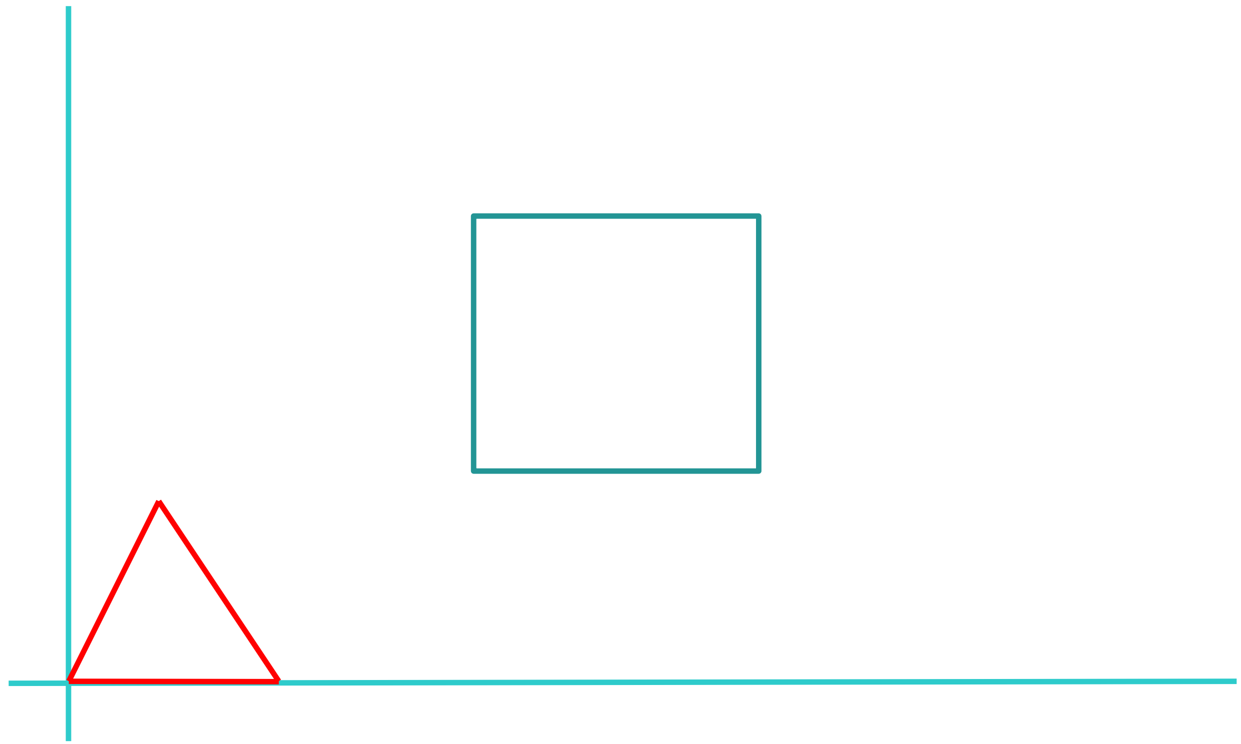
- if P and R are convex polygons with n and m edges, then $P \oplus R$ has at most $n+m$ edges.

Minkowski Sum

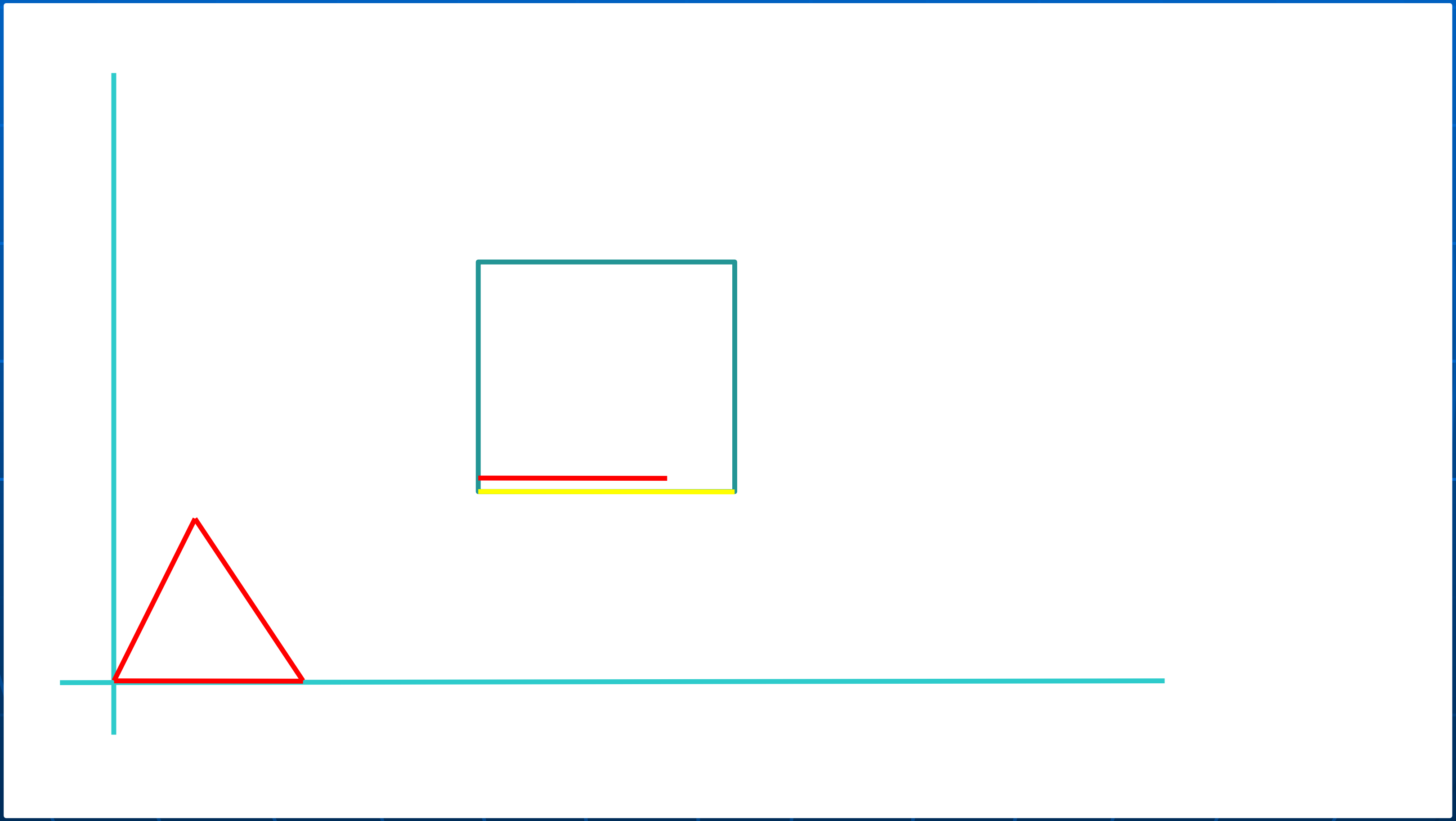
- O(n+m)-time Algorithm

1. $i \leftarrow 1$; $j \leftarrow 1$
2. $V_{n+1} \leftarrow V_1$; $W_{m+1} \leftarrow W_1$
3. repeat
4. Add $V_i + W_j$ as a vertex to $P \oplus R$
5. if angle(V_iV_{i+1}) < angle(W_jW_{j+1})
6. then $i \leftarrow (i+1)$
7. else if angle(V_iV_{i+1}) > angle(W_jW_{j+1})
8. then $j \leftarrow (j+1)$
9. else $i \leftarrow (i+1)$
10. $j \leftarrow (j+1)$
11. until $i = n + 1$ and $j = m + 1$

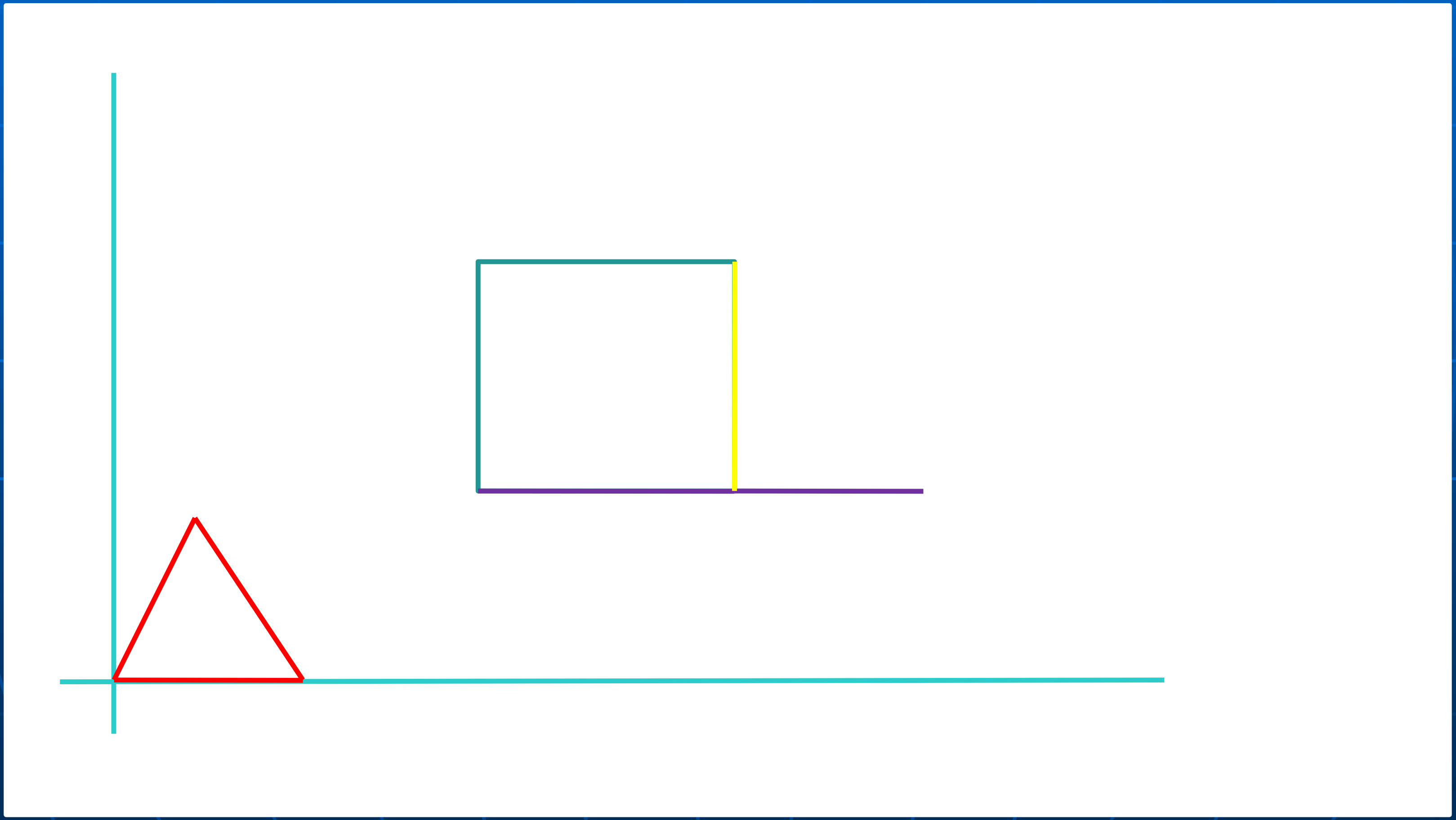
Minkowski Sum



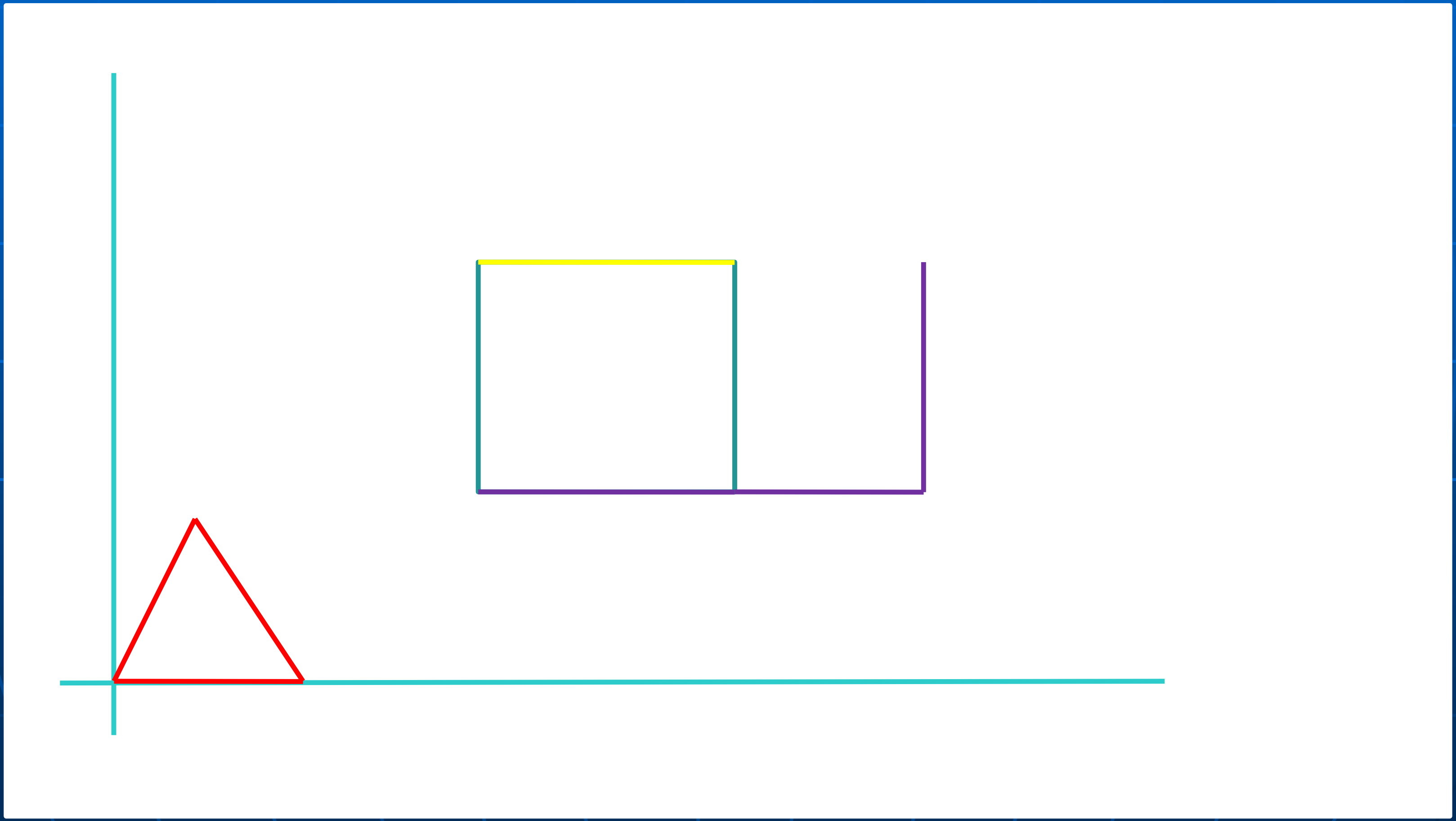
Minkowski Sum



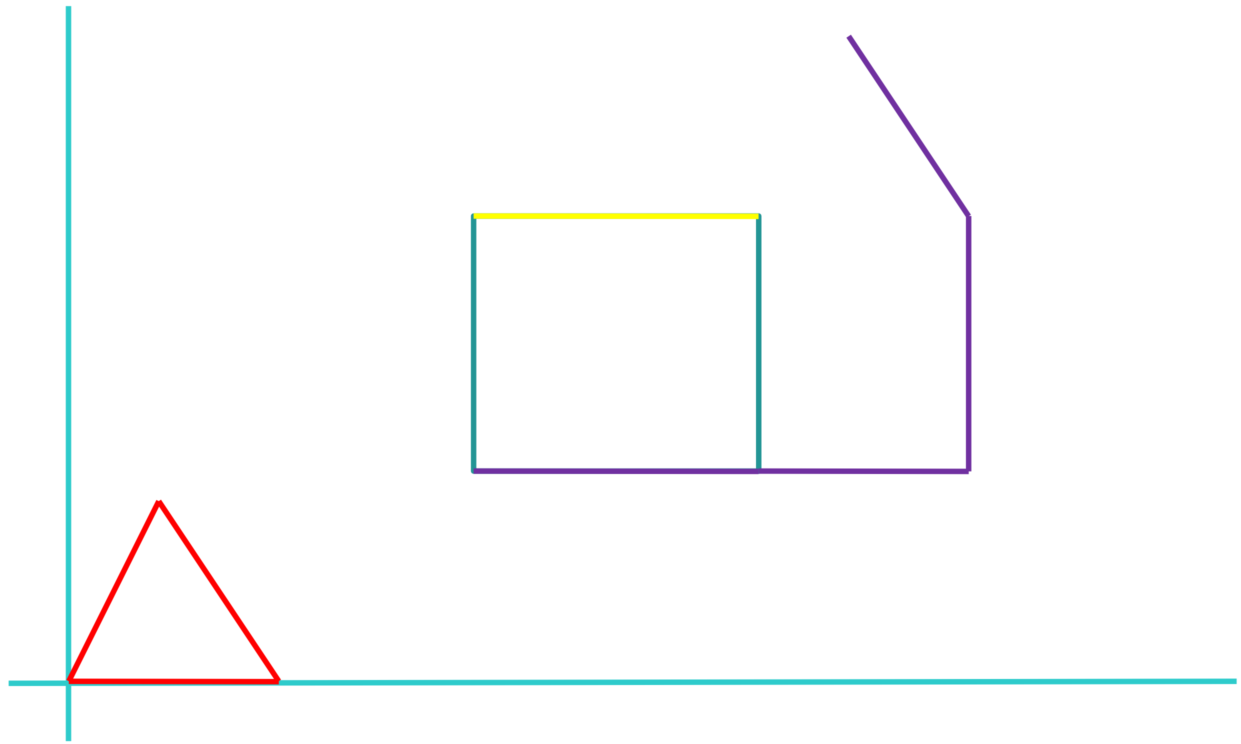
Minkowski Sum



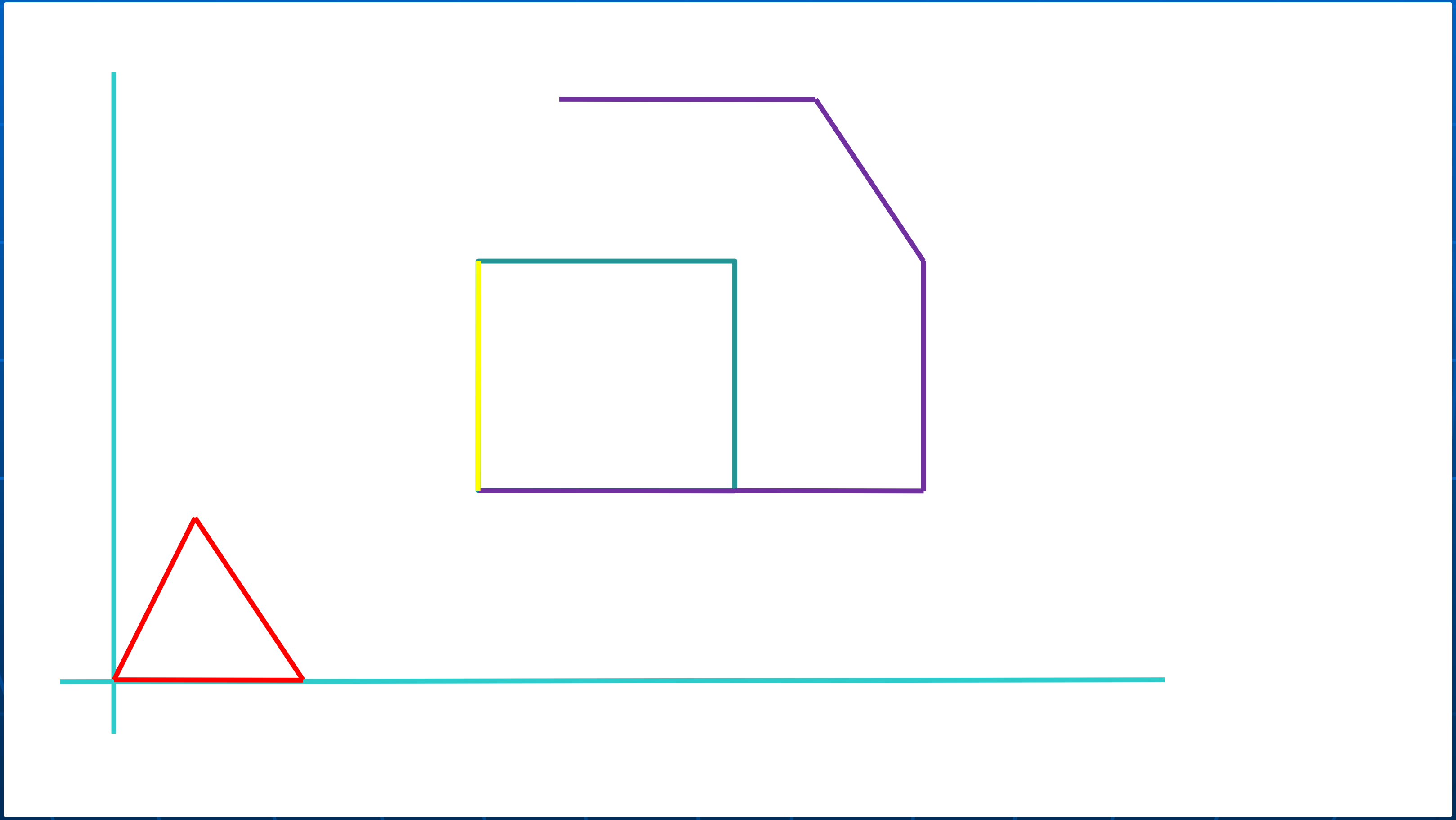
Minkowski Sum



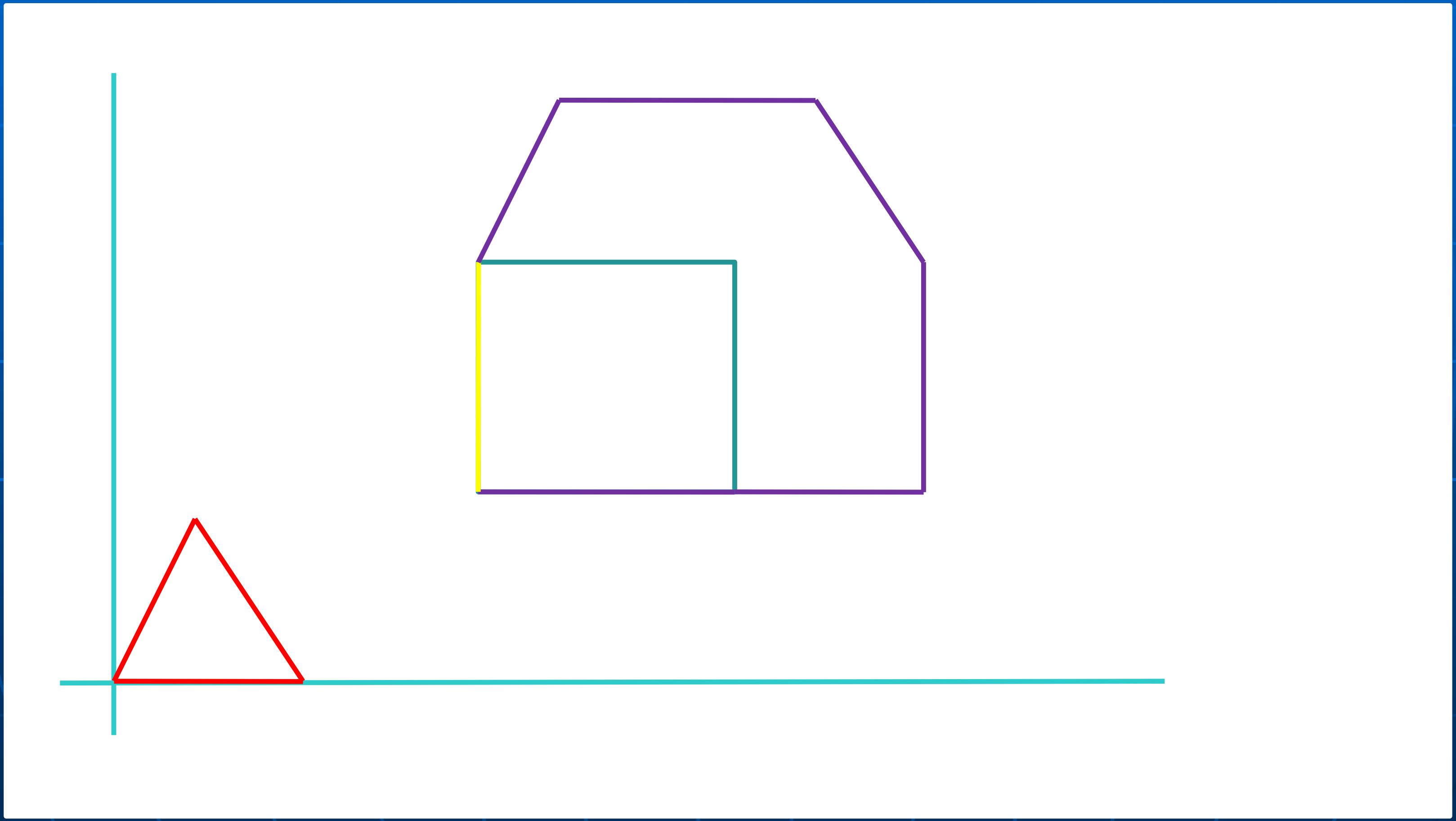
Minkowski Sum



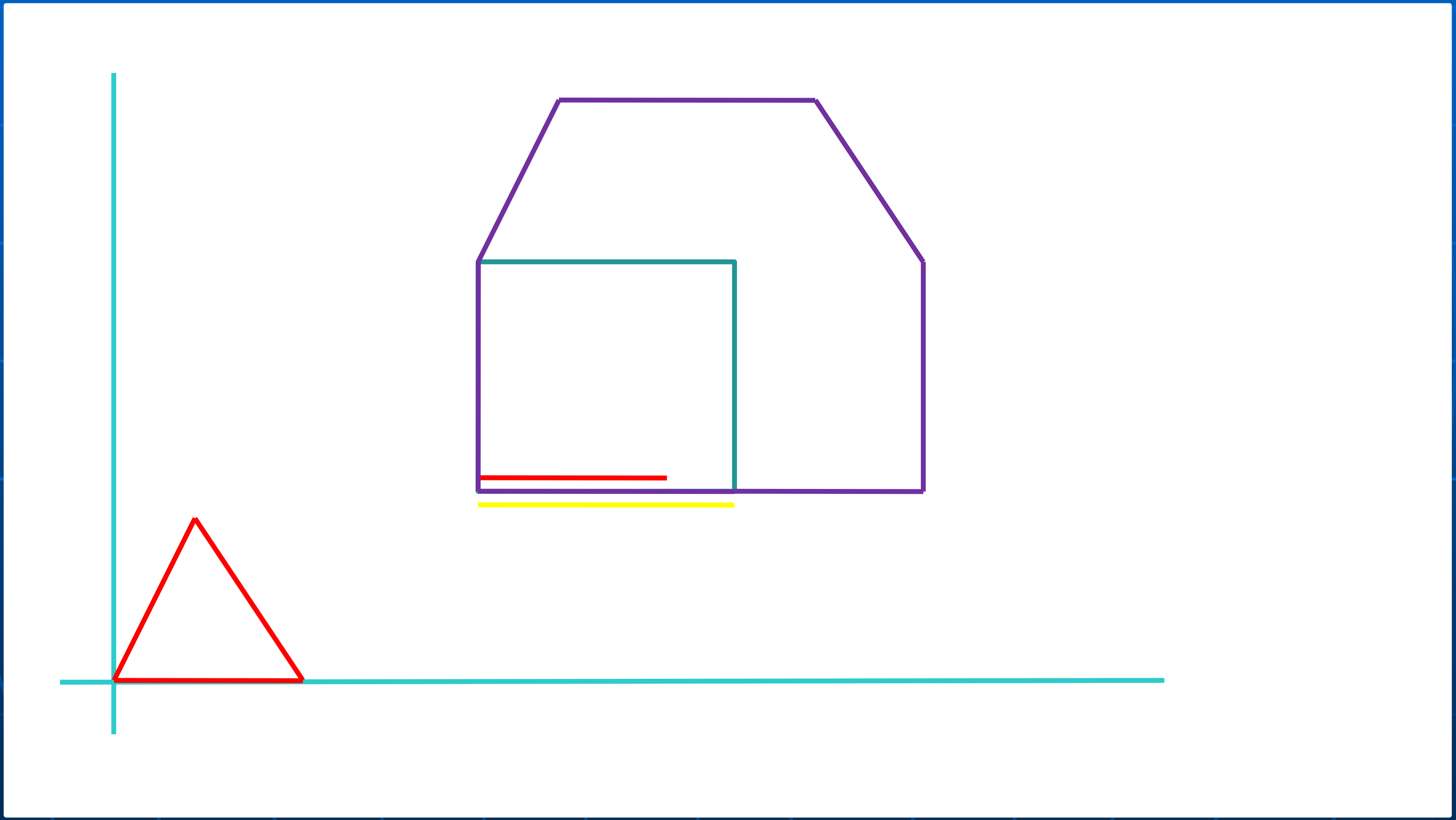
Minkowski Sum



Minkowski Sum

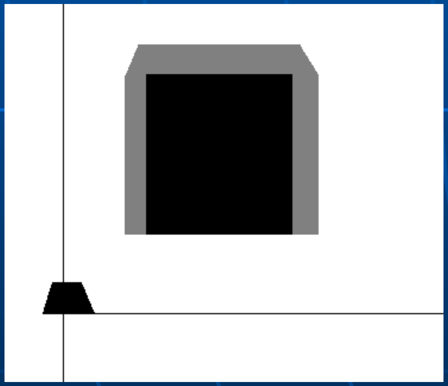


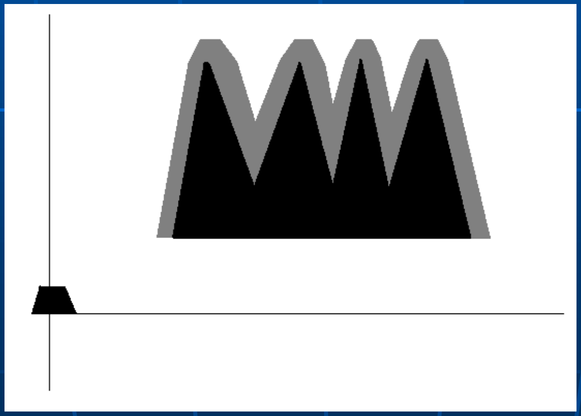
Minkowski Sum

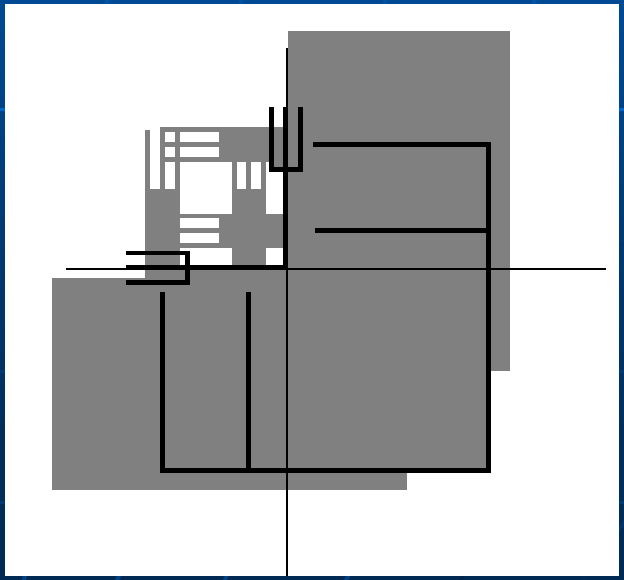


Minkowski Sum

- convex + convex : $O(n+m)$
- convex + non-convex : $O(nm)$
- non-convex + non-convex : $O(n^2m^2)$

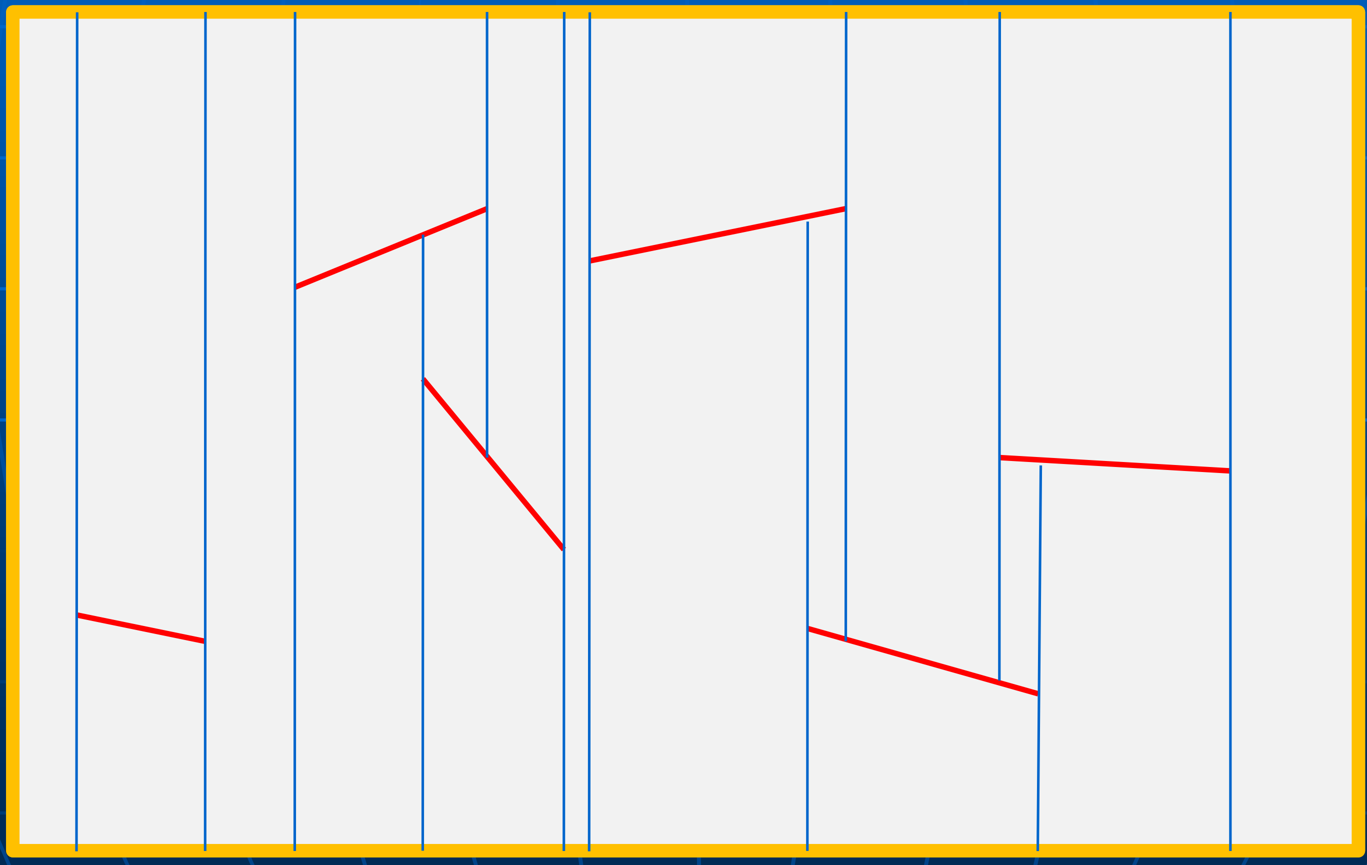






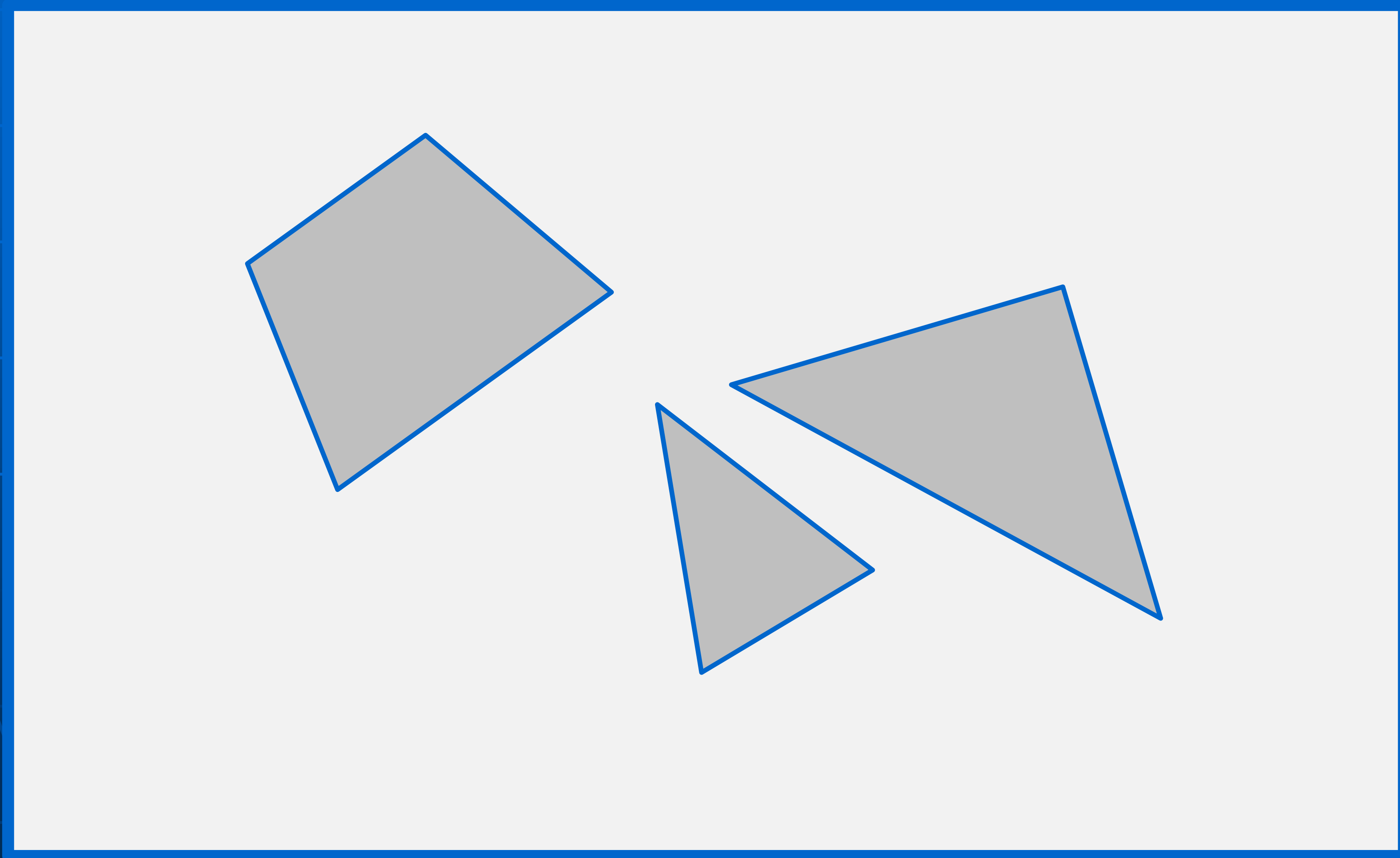
Trapezoidal Map

Merit: Trapezoids info. + their adjacency info.
O(n log n) time, O(n) space



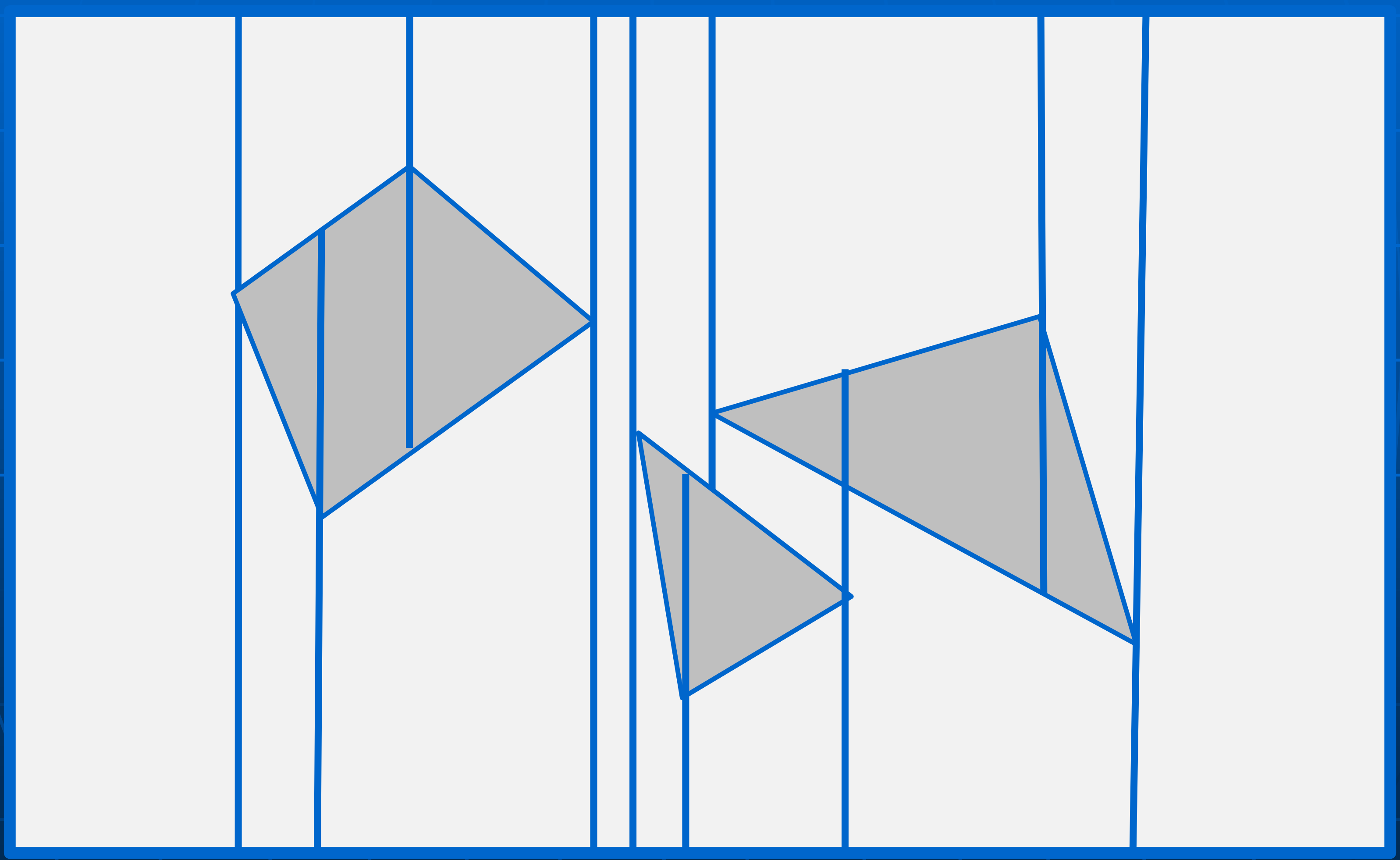
Build Road map of Free Space

1. Let E be the set of edge of C-obstacles.



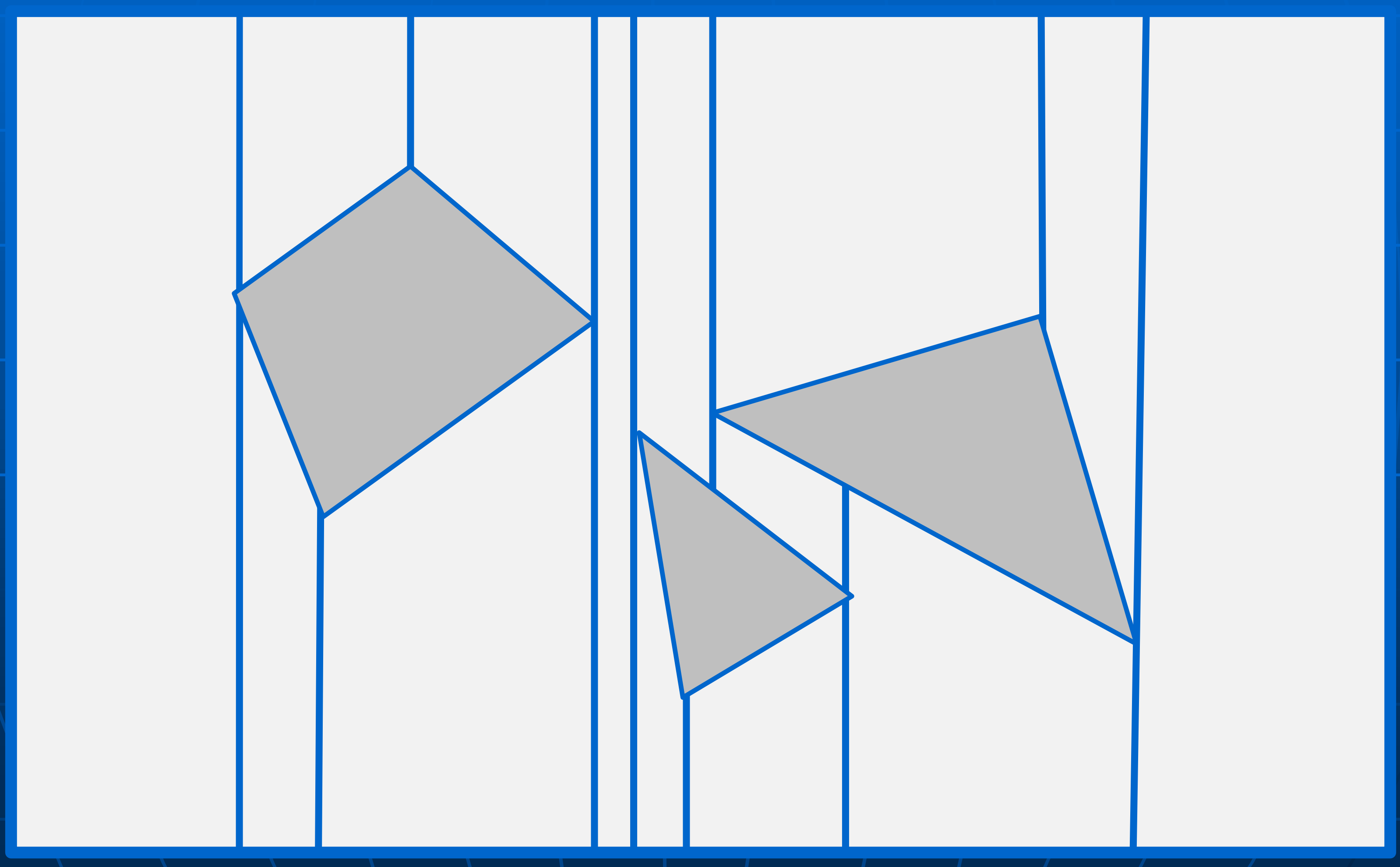
Build Road map of Free Space

2. Compute trapezoidal map of E.



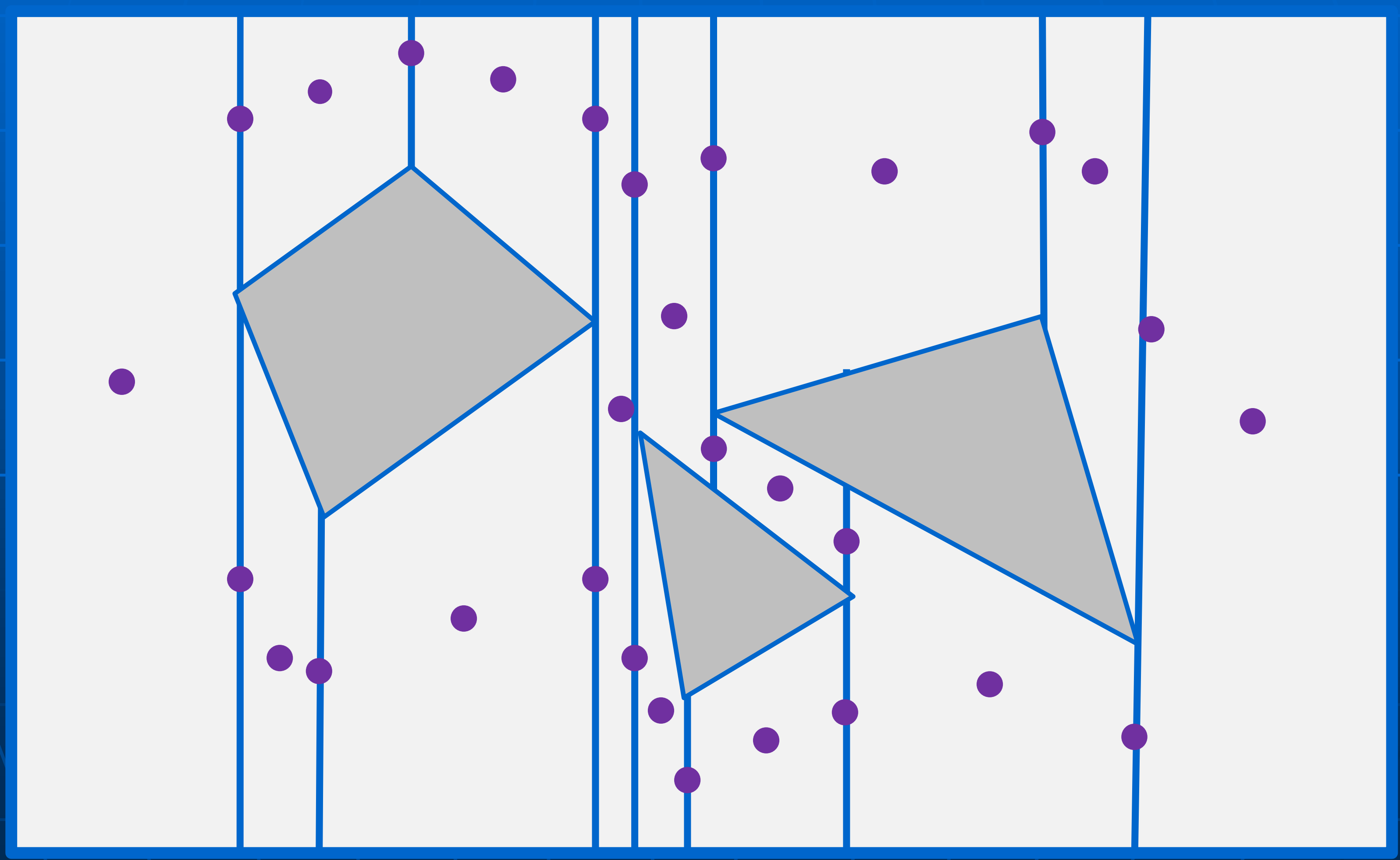
Build Road map of Free Space

3. Remove trapezoids lying inside the C-obstacles



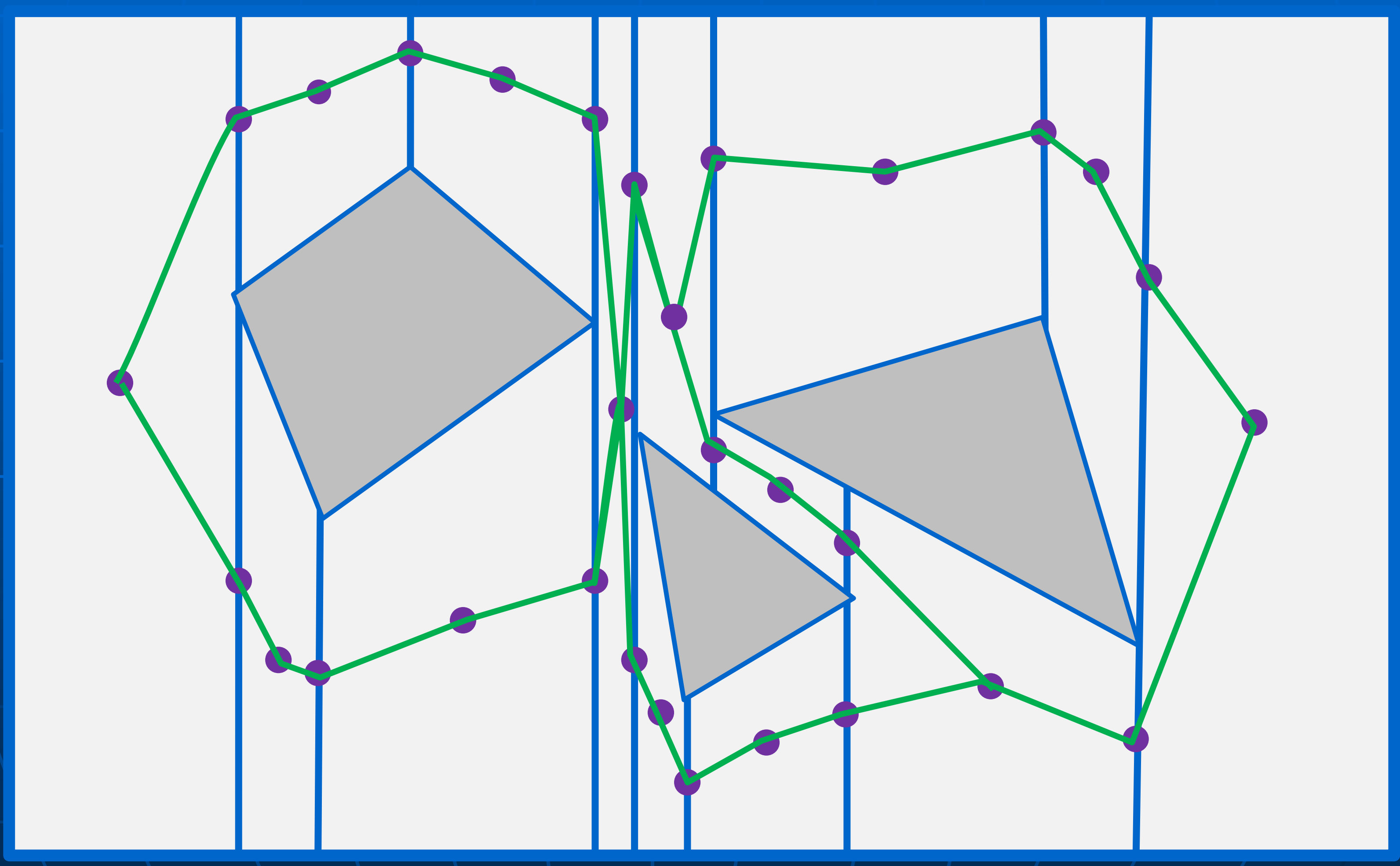
Build Road map of Free Space

4. Generate trapezoid centers and mid-points of verticals

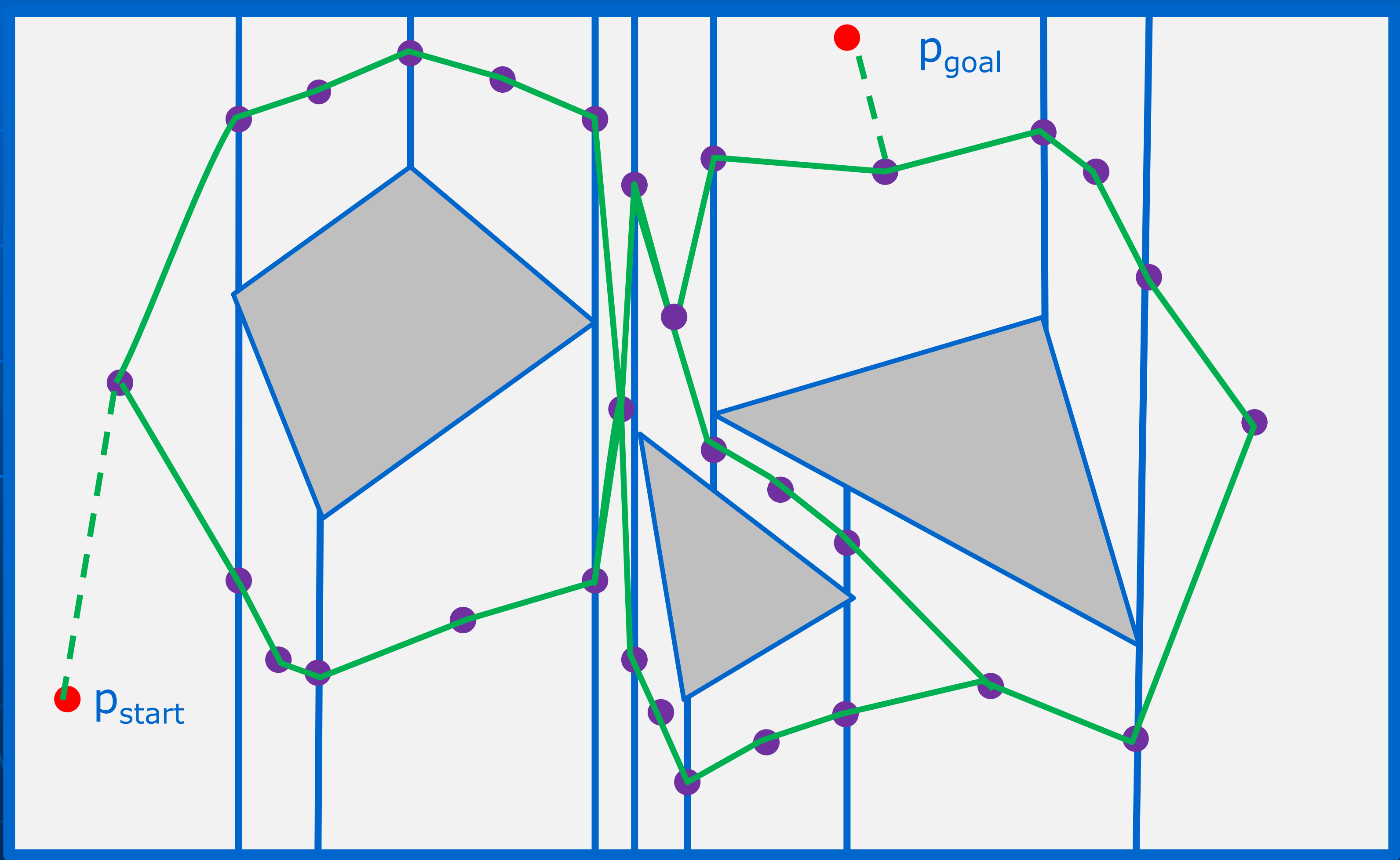


Build Road map of Free Space

5. Complete the road map.



Compute Robot Path from p_{start} to p_{goal}



Rotation-Allowed Robot

- C-obstacles in Configuration space

$CP_i := \{(x,y,\phi) \in R^2 \times [0:360) : R(x,y,\phi) \cap P_i \neq 0\}$