GEOMETRIC ALGORITHMS

Huỳnh Thị Mỹ Duyên – 19521438

Phan Tung – 19522495

Lương Tường Quy – 19522108

Huỳnh Kim Phát – 19521992

GEOMETRIC ALGORITHMS

- Primitive operations
- Hough Line Transform
- Convex hull
- Closest pair
- Voronoi diagram

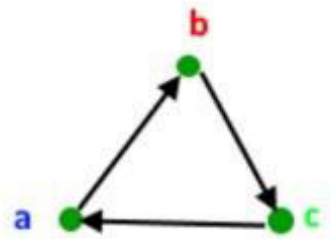
GEOMETRIC ALGORITHMS

- Primitive operations
- Hough Line Transform
- Convex hull
- Closest pair
- Voronoi diagram

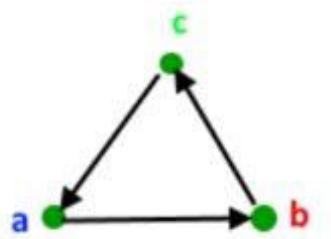
PRIMITIVE OPERATIONS

- Counter Clockwise Turns
- Line Intersection
- Point lies inside or outside a polygon

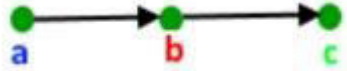
Counter Clockwise Turns



Clockwise



Counterclockwise

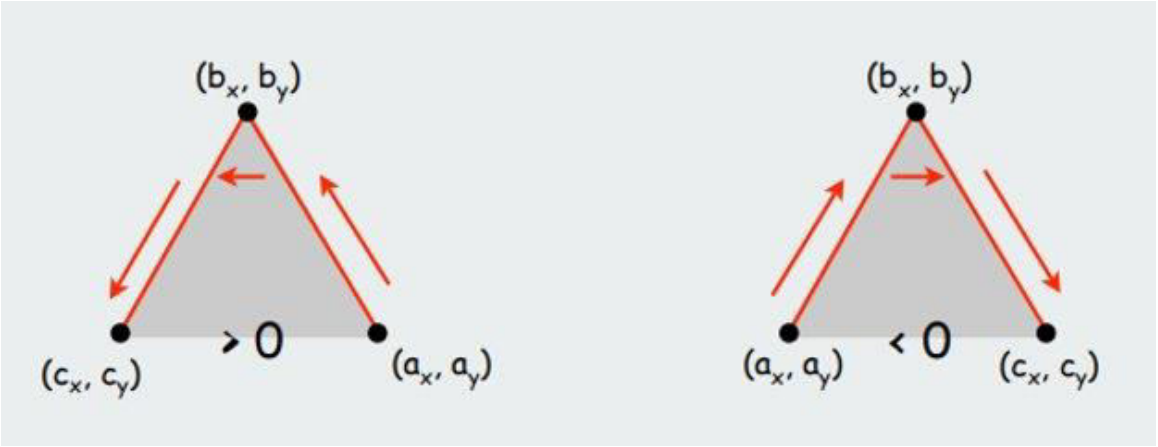


Collinear

Counter Clockwise Turns

- Nếu Area > 0 thì a-b-c là counterclockwise.
- Nếu Area < 0 thì a-b-c là clockwise.
- Nếu Area = 0 thì a-b-c là collinear.

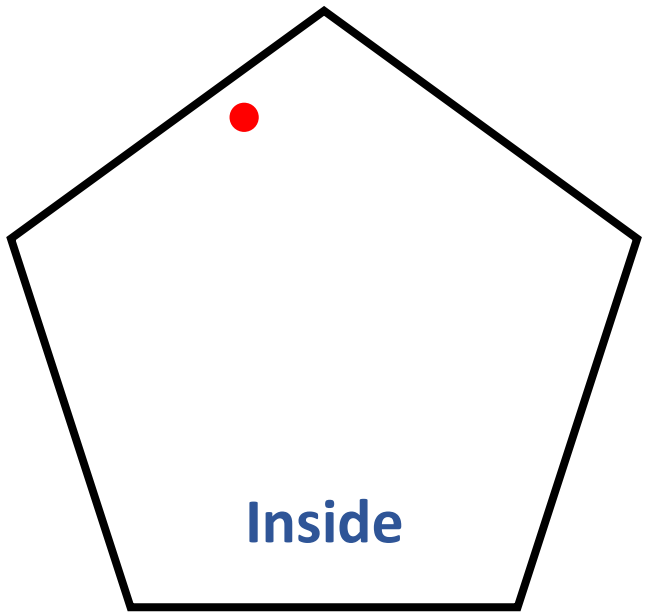
$$2 \times Area(a,b,c) = \begin{vmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{vmatrix} = (b_x - a_x)(c_y - a_y) - (b_y - a_y)(c_x - a_x)$$

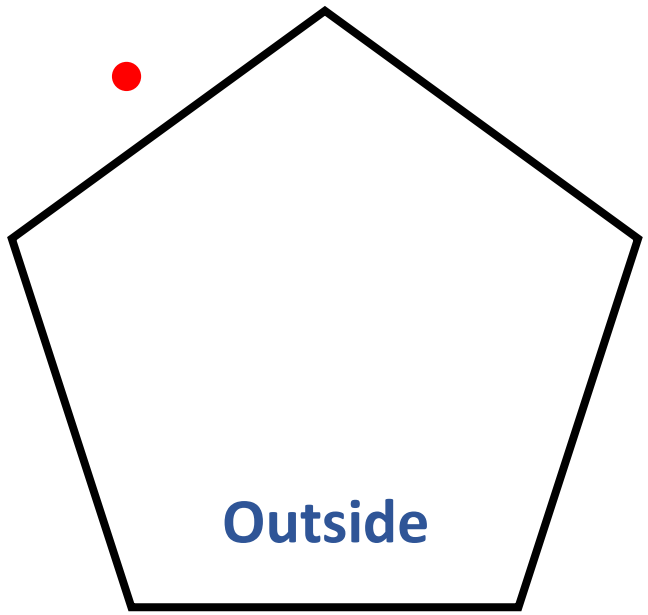


PRIMITIVE OPERATIONS

- Counter Clockwise Turns
- Line Intersection
- Point lies inside or outside a polygon

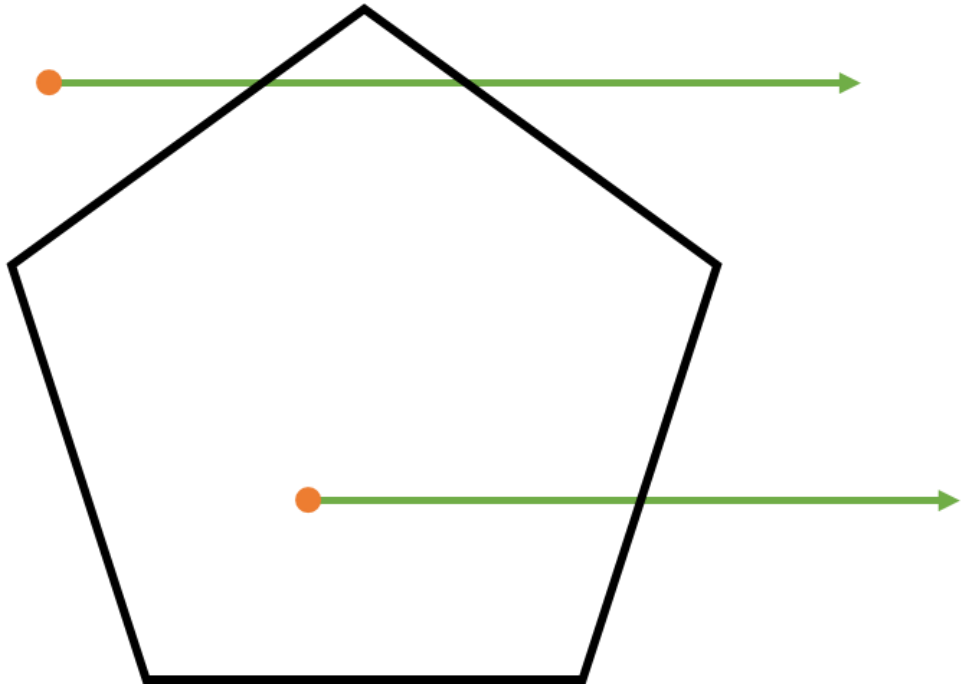
Point Lies Inside or Outside a Polygon





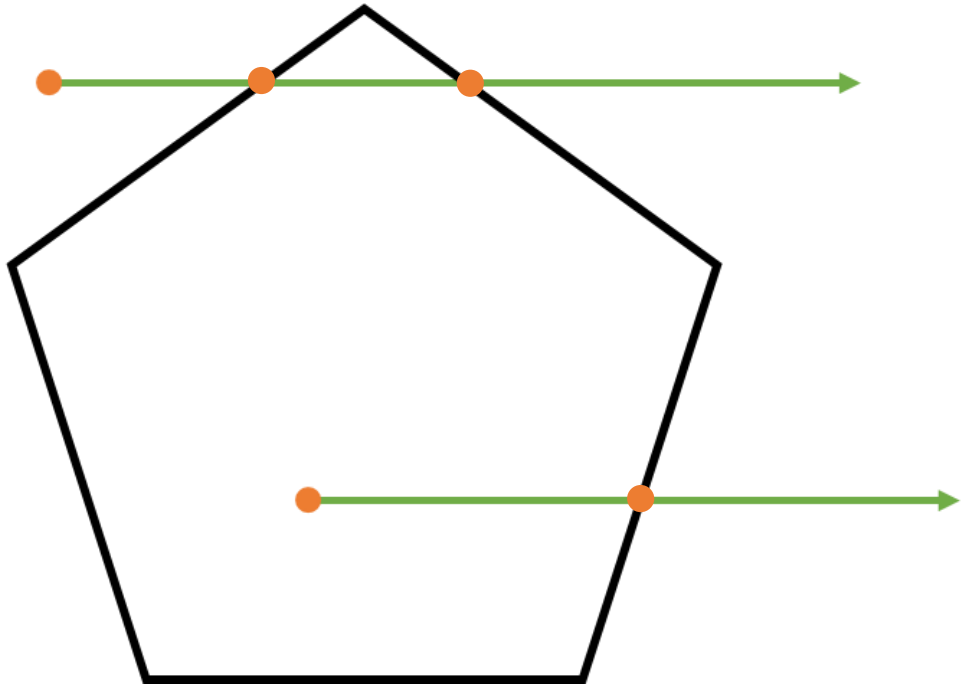
Point Lies Inside or Outside a Polygon

Bước 1: Vẽ một đường ngang ở bên phải của mỗi điểm và kéo dài nó đến vô cùng



Point Lies Inside or Outside a Polygon

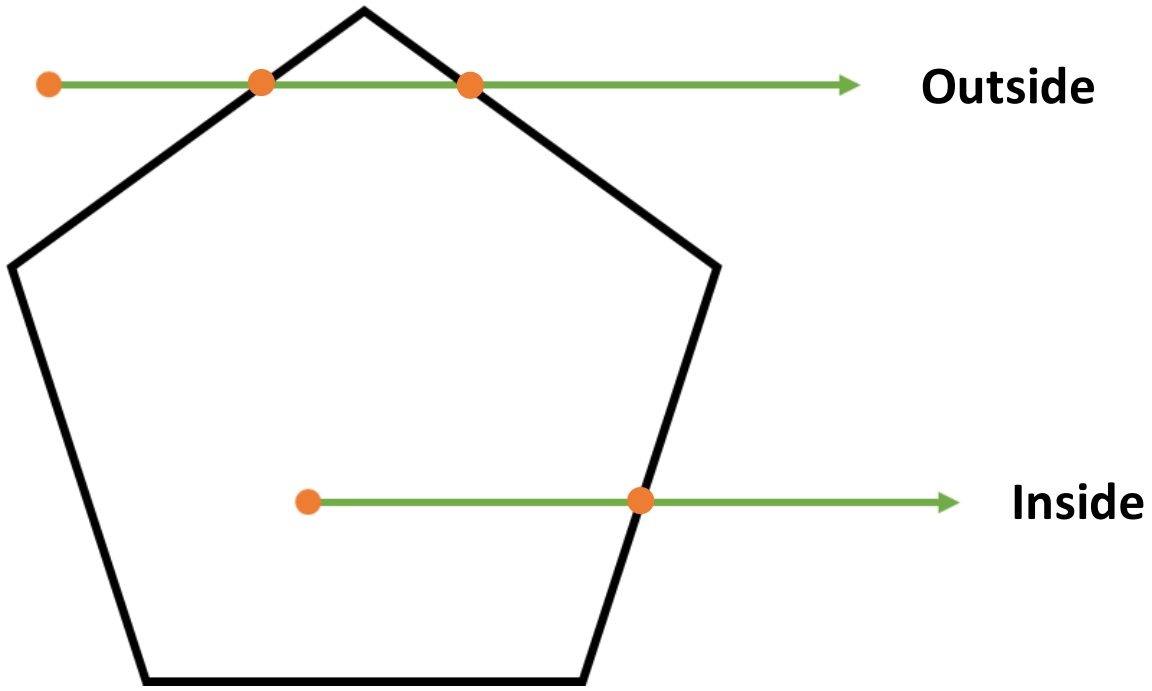
Bước 2: Đếm số lần giao điểm của đường thẳng với các cạnh đa giác.



Point Lies Inside or Outside a Polygon

Bước 3: Kiểm tra nếu số giao điểm là:

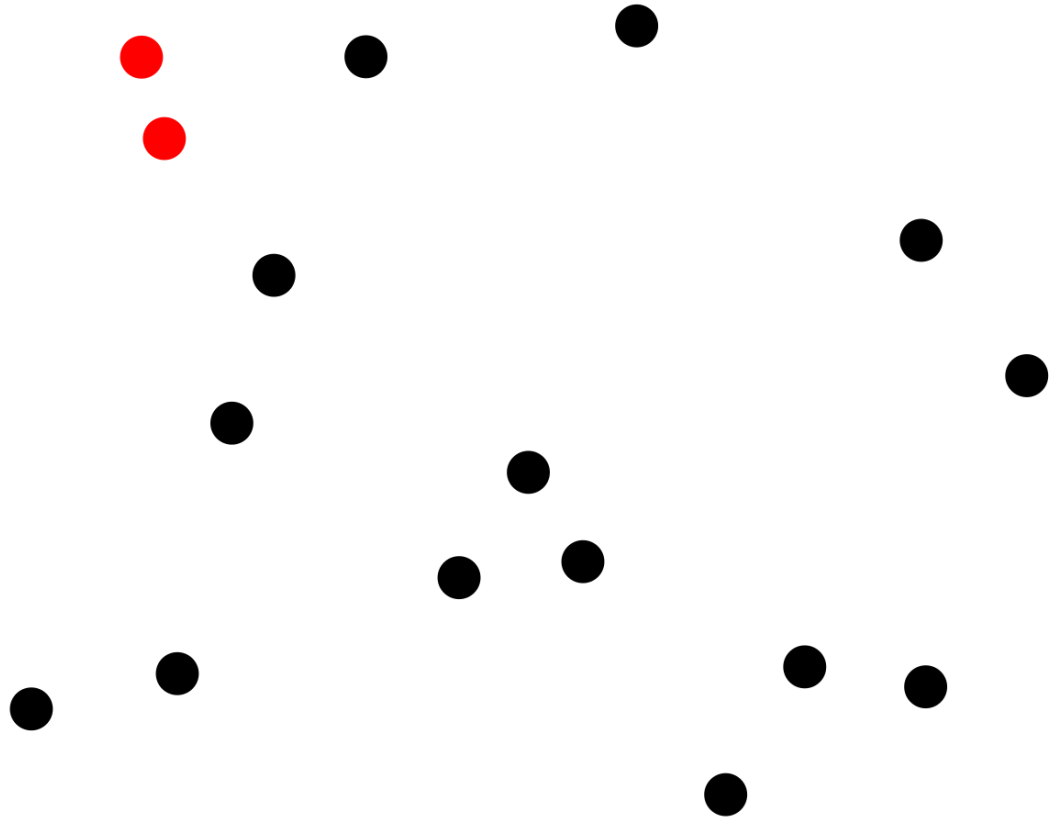
- **Số lẻ → điểm ban đầu nằm bên trong (hoặc trên) đa giác.**
- **Số chẵn (hoặc = 0) → điểm ban đầu nằm bên ngoài đa giác.**



GEOMETRIC ALGORITHMS

- Primitive operations
- Hough Line Transform
- Convex hull
- Closest pair
- Voronoi diagram

Closest-pair



Closest-pair

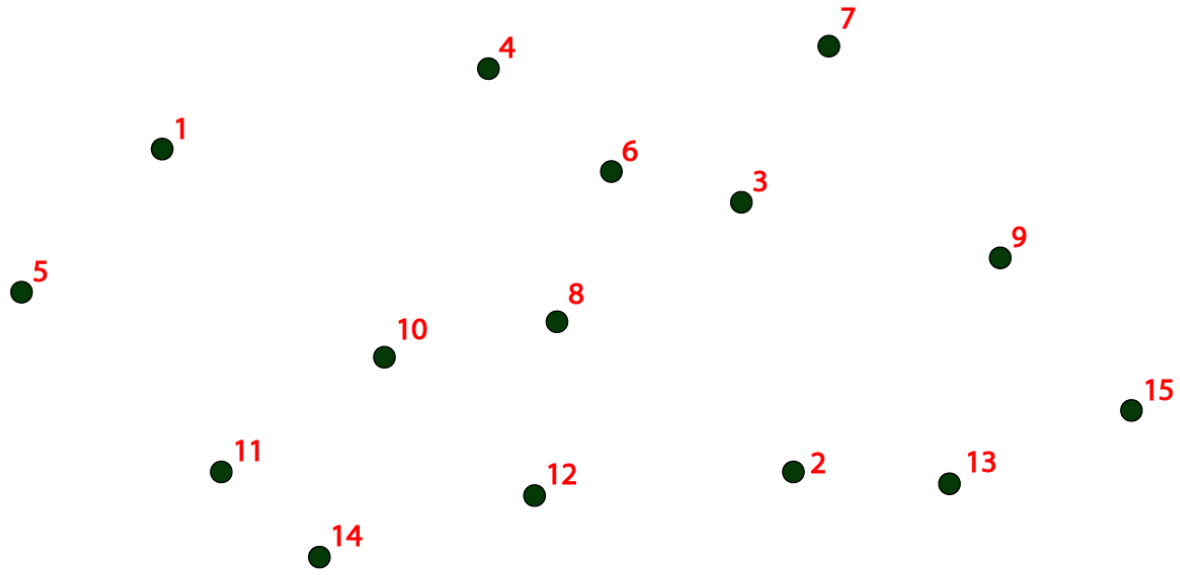
- Khoảng cách giữa 2 điểm p và q trong không gian 2 chiều.
- Công thức:

$$d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$$

Closest-pair

Input: Một mảng n điểm P[]

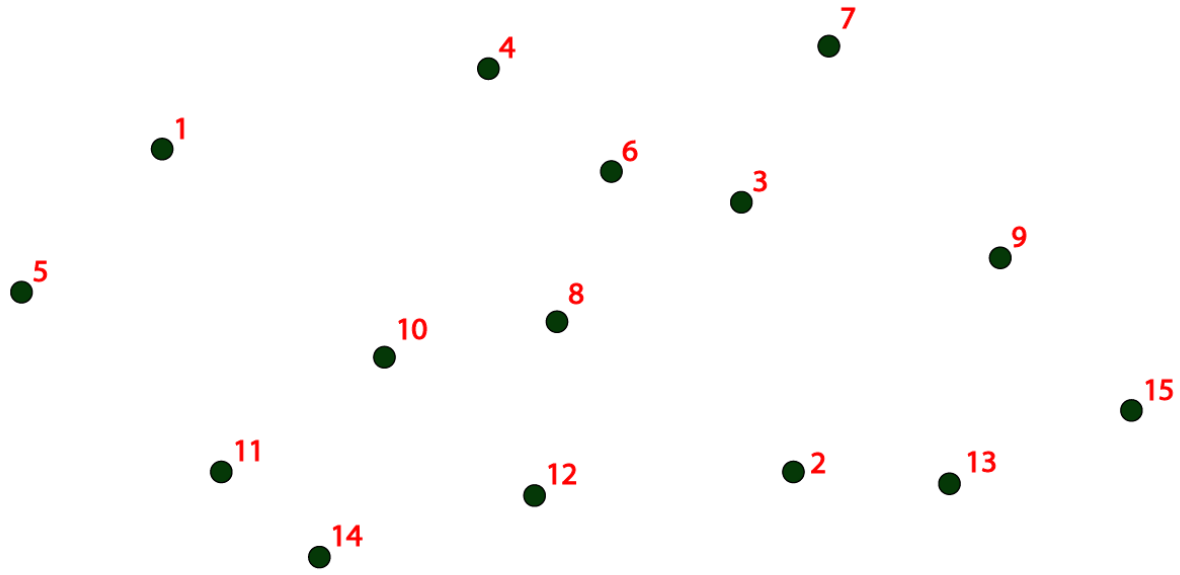
P = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}



Closest-pair

Bước 1: Sắp xếp tất cả các điểm theo tọa độ x

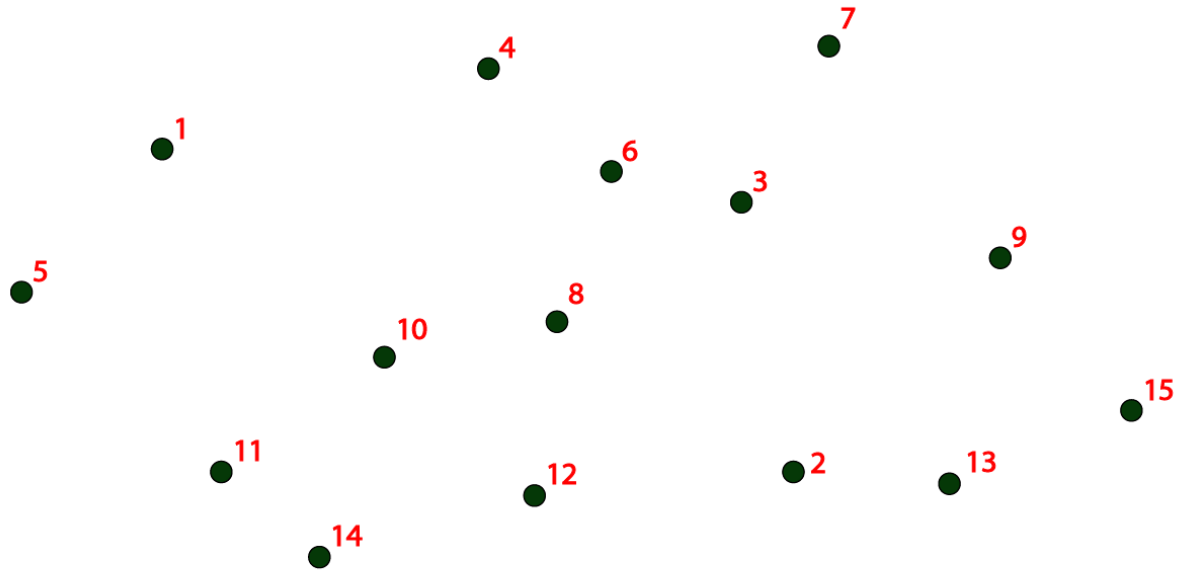
P = {5, 1, 11, 14, 10, 4, 12, 8, 6, 3, 2, 7, 13, 9, 15}



Closest-pair

Bước 2: Chia tất cả các điểm thành hai nửa

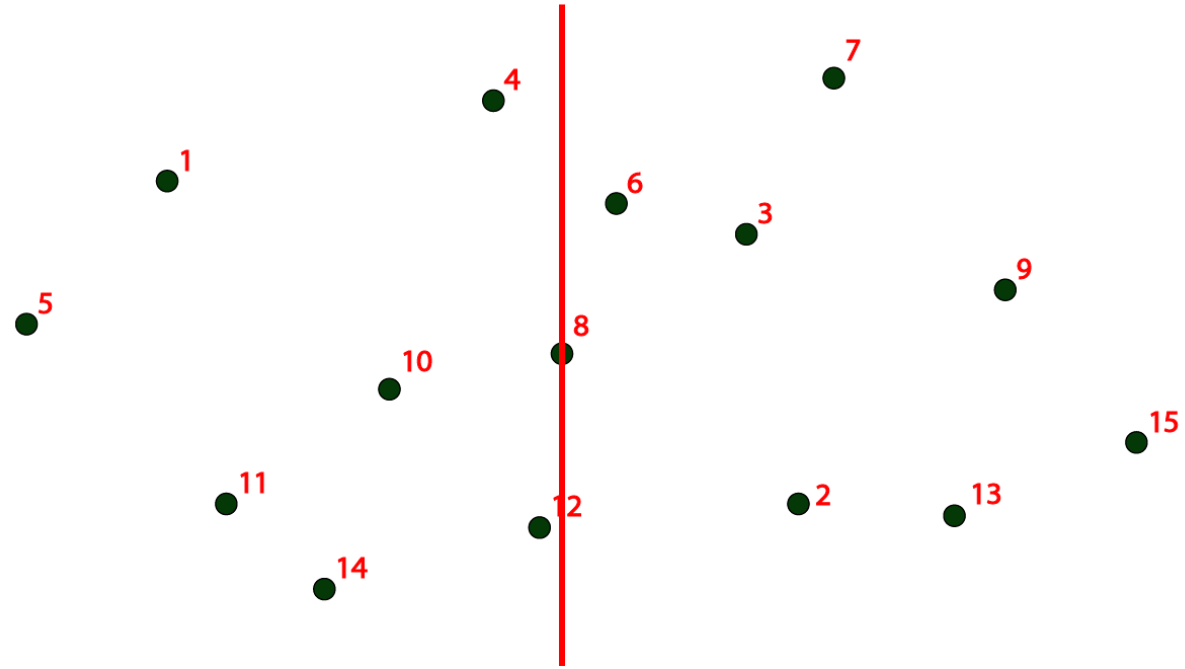
P = {5, 1, 11, 14, 10, 4, 12, 8, 6, 3, 2, 7, 13, 9, 15}



Closest-pair

Bước 2: Chia tất cả các điểm thành hai nửa

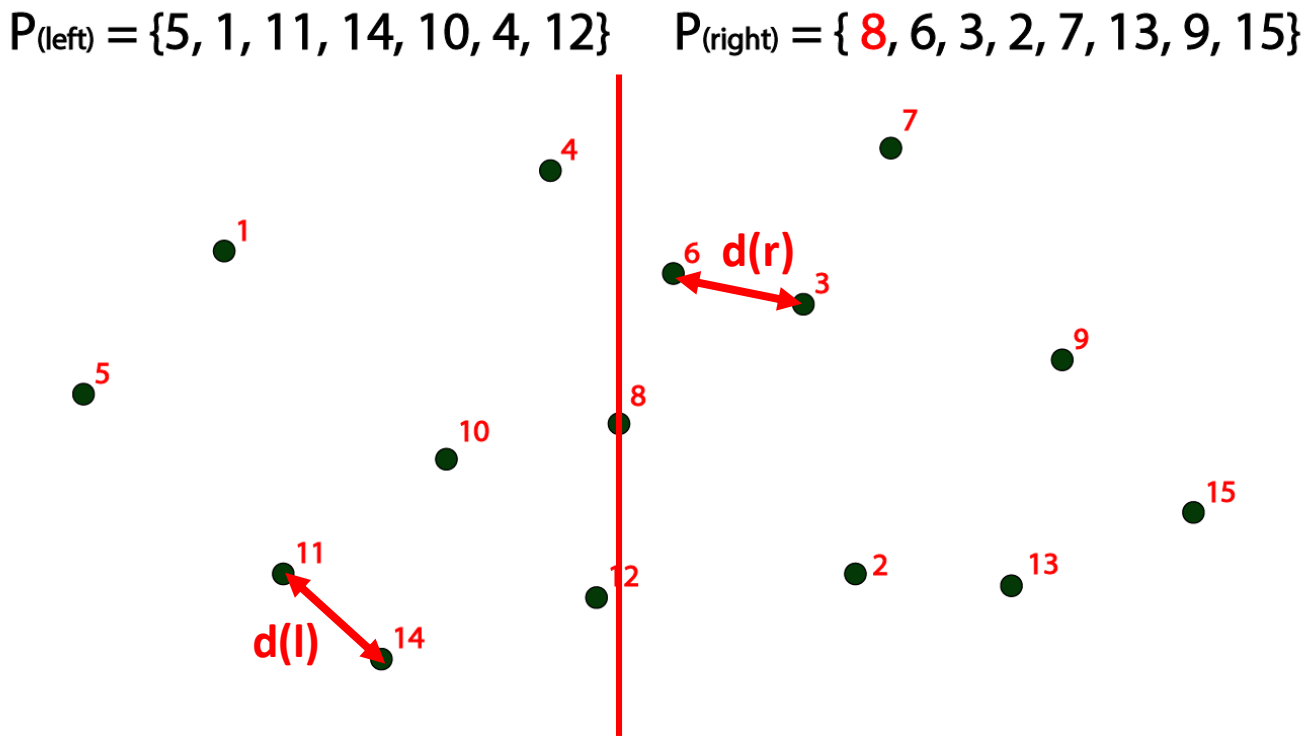
$P_{(left)} = \{5, 1, 11, 14, 10, 4, 12\}$ $P_{(right)} = \{ 8, 6, 3, 2, 7, 13, 9, 15\}$



Closest-pair

Bước 3:

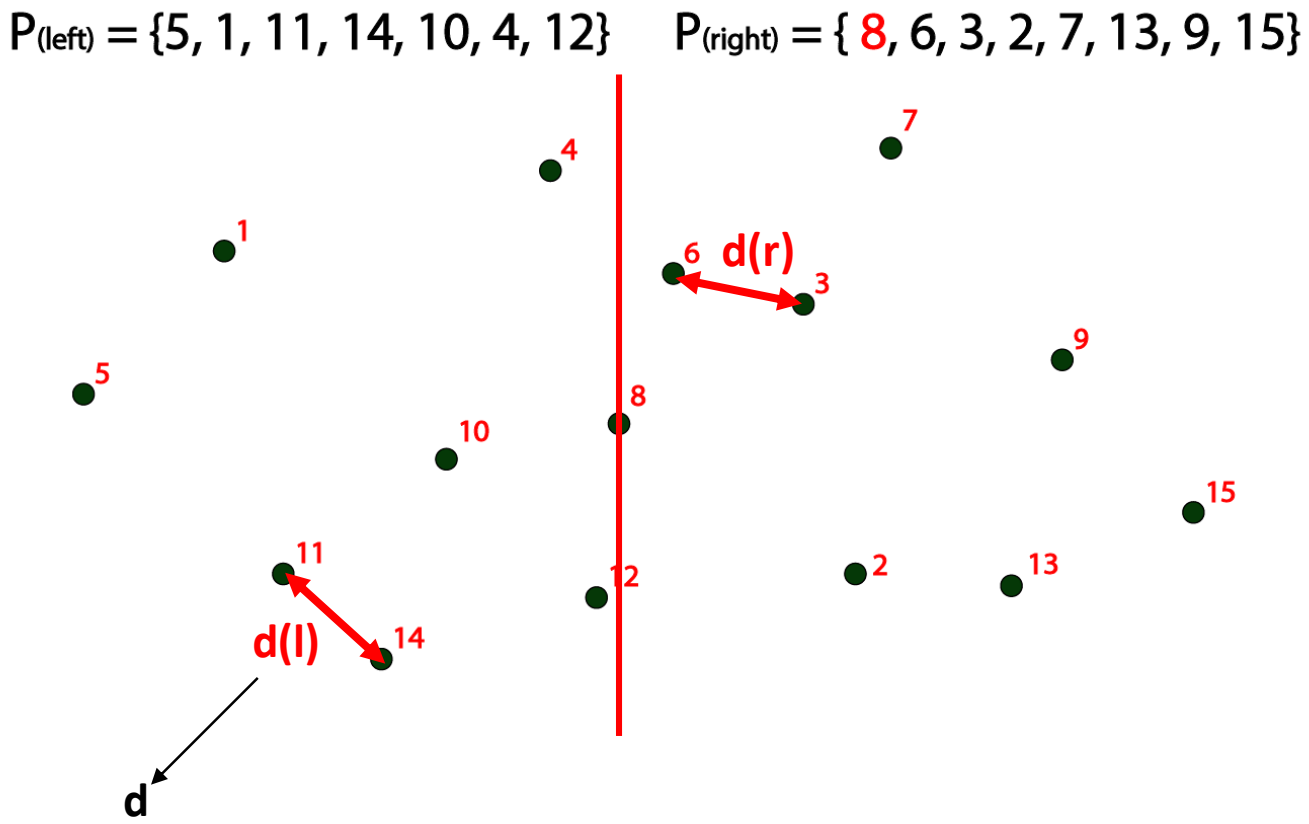
- **Tìm d(l) và d(r) lần lượt là khoảng cách giữa 2 điểm gần nhất trong 2 tập P(left) và P(right)**



Closest-pair

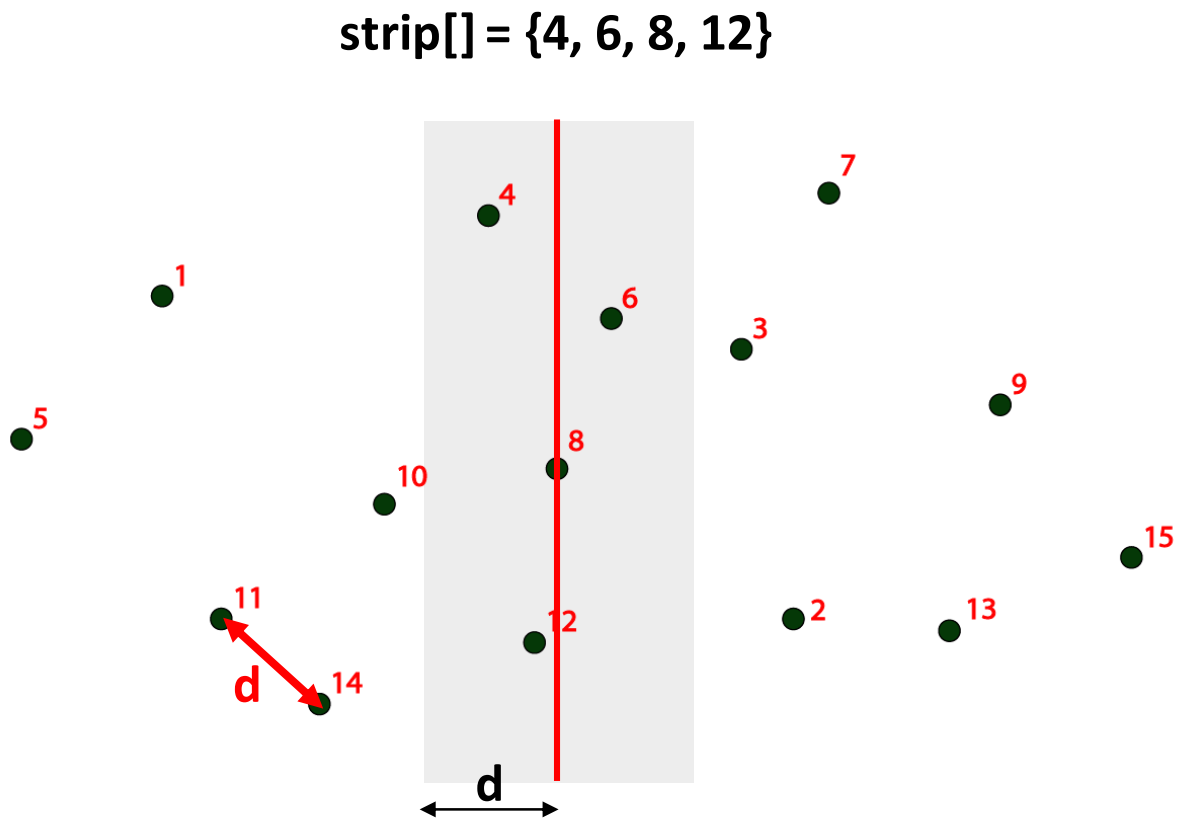
Bước 3:

- **Tìm d(l) và d(r) lần lượt là khoảng cách giữa 2 điểm gần nhất trong 2 tập P(left) và P(right)**
- **Tìm d = min(d(l), d(r))**



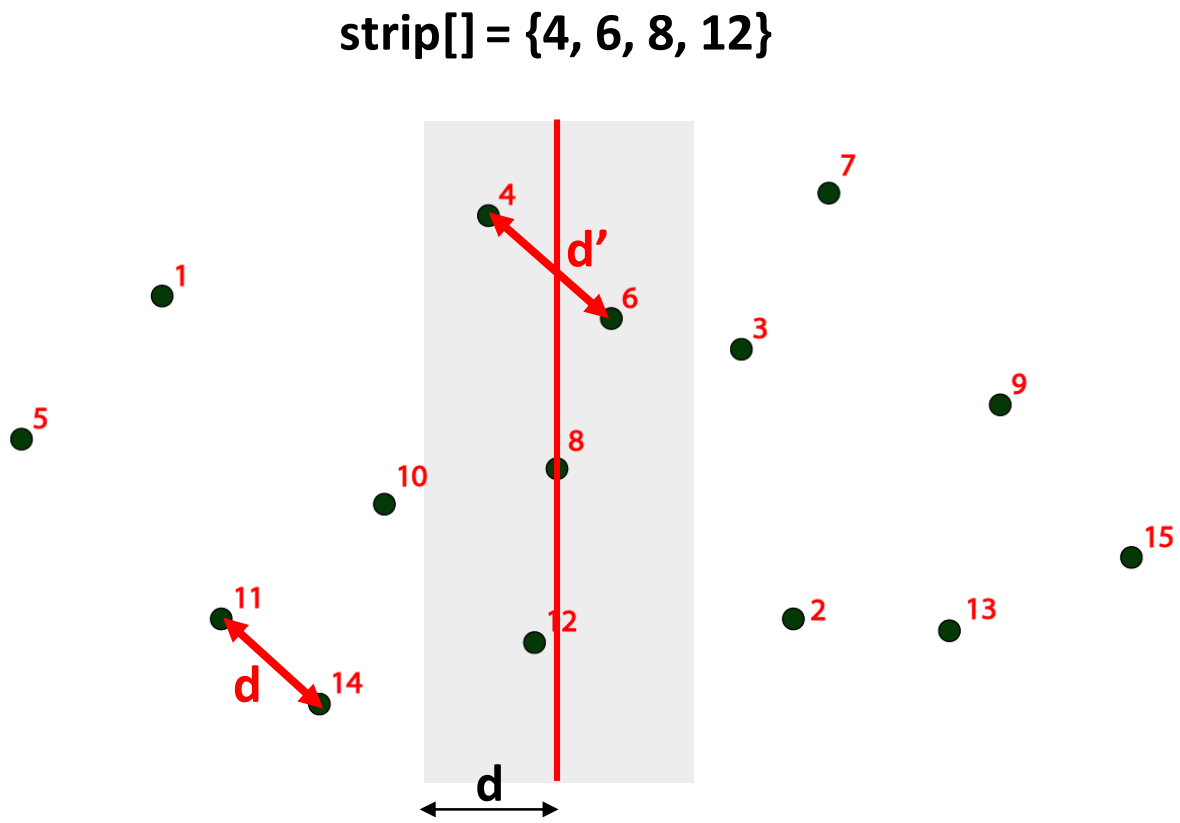
Closest-pair

Bước 4: Tạo một dải mảng [] lưu trữ tất cả các điểm cách đường giữa chia hai tập hợp nhiều nhất là d.



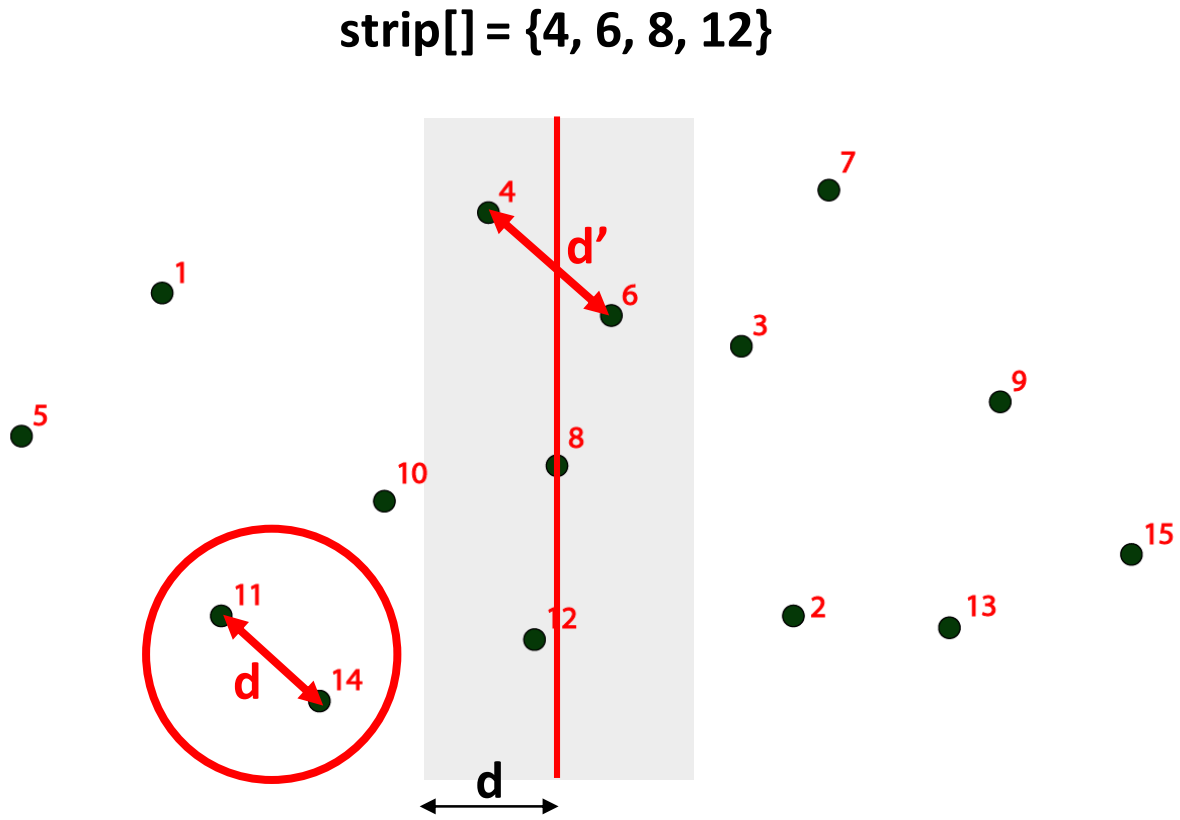
Closest-pair

Bước 5: Tìm khoảng cách d’ giữa 2 điểm gần nhất trong strip[]



Closest-pair

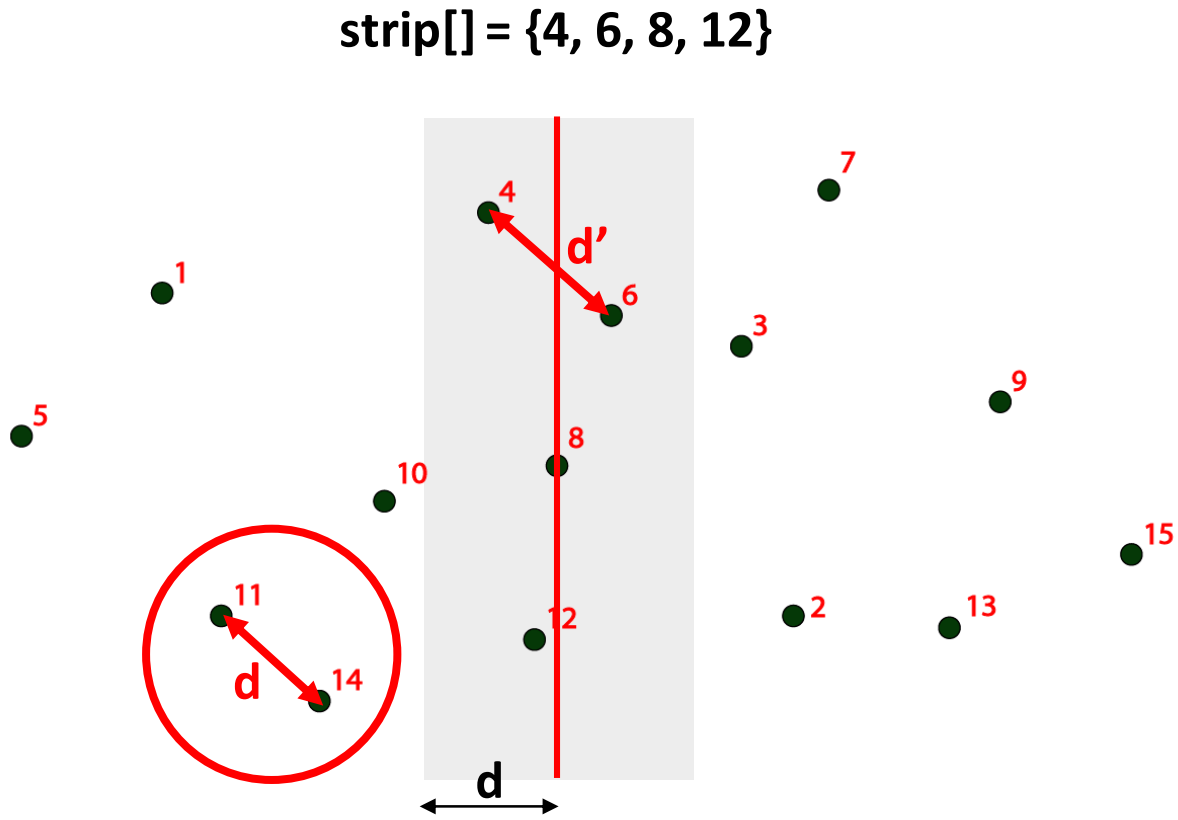
Output: min(d, d')



Closest-pair

Output: min(d, d')

⇒ Độ phức tạp theo thuật toán Divide and Conquer: $O(n * (\log n)^2)$



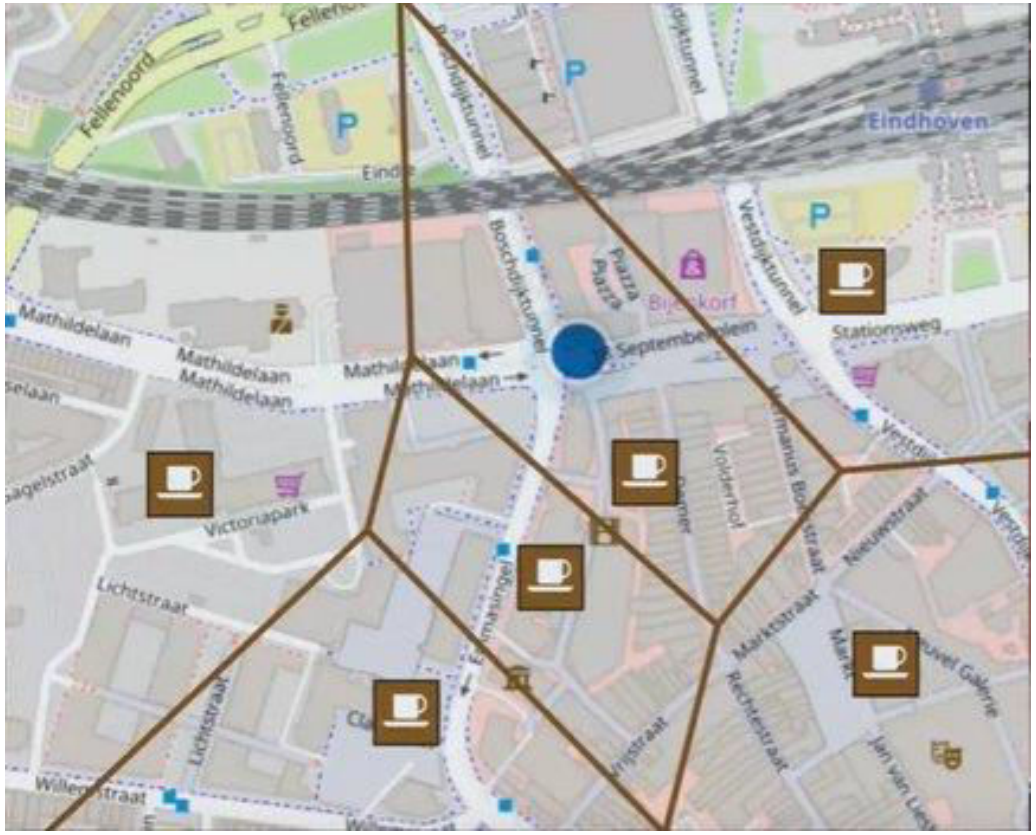
Applications

- Dynamic minimum spanning trees
- Two-optimization heuristics in combinatorial optimization
- Straight skeletons and roof design
- Ray-intersection diagram
- Other collision detection applications
- Hierarchical clustering
- Traveling salesman heuristics
- Greedy matching
- Constructive induction
- …

GEOMETRIC ALGORITHMS

- Primitive operations
- Hough Line Transform
- Convex hull
- Closest pair
- Voronoi diagram

Voronoi Diagram



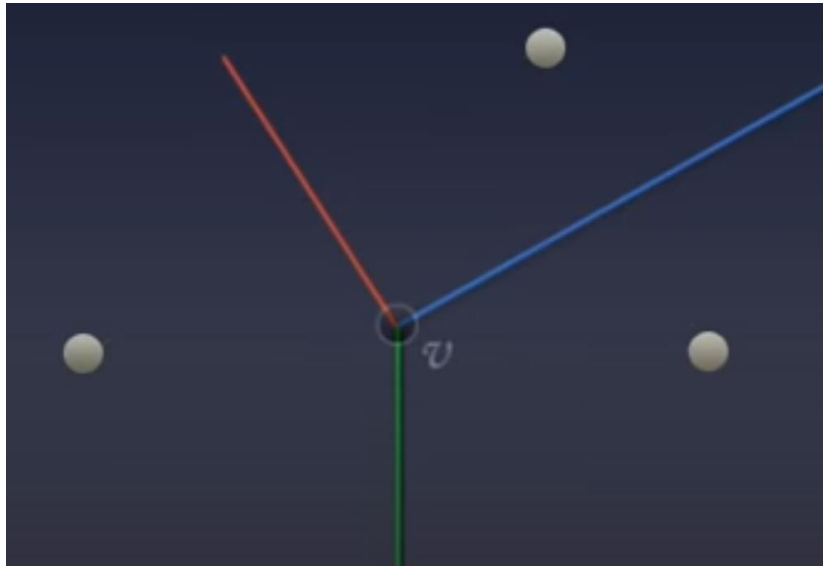
Voronoi Diagram

- **Voronoi Edge (cạnh Voronoi)**

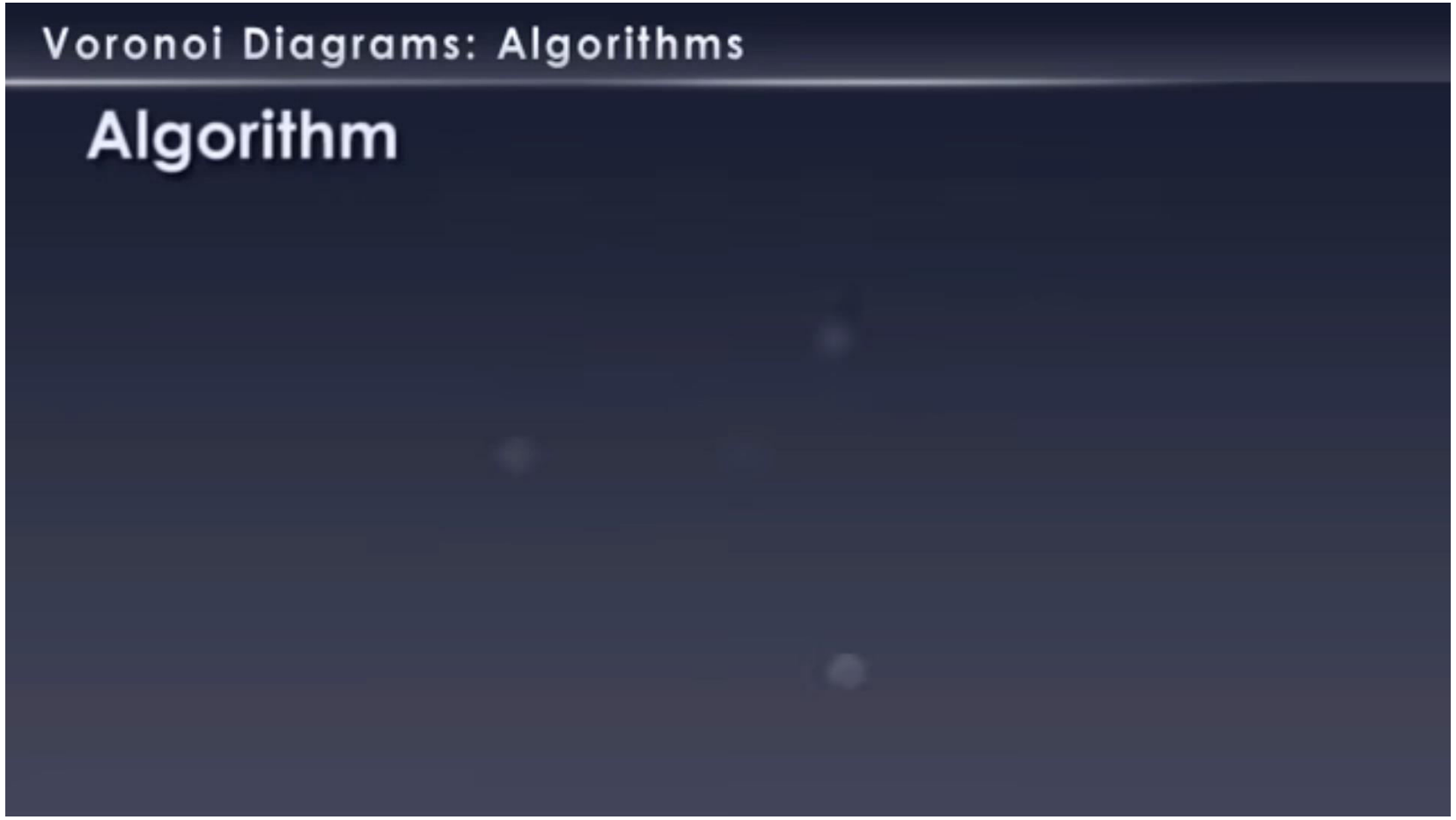
Voronoi Diagram

- **Voronoi Edge (cạnh Voronoi)**

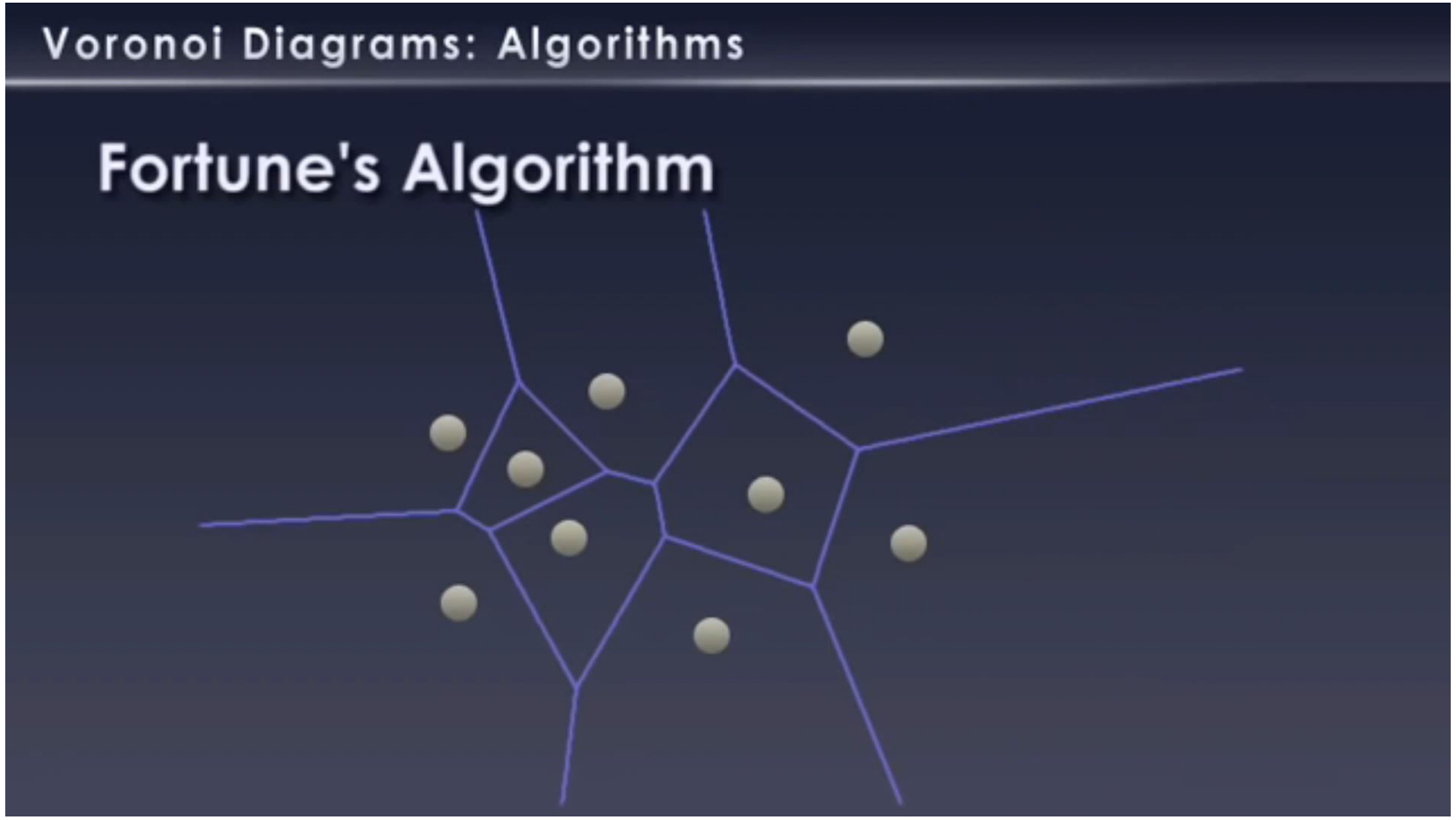
- **Voronoi vertex (đỉnh Voronoi)**



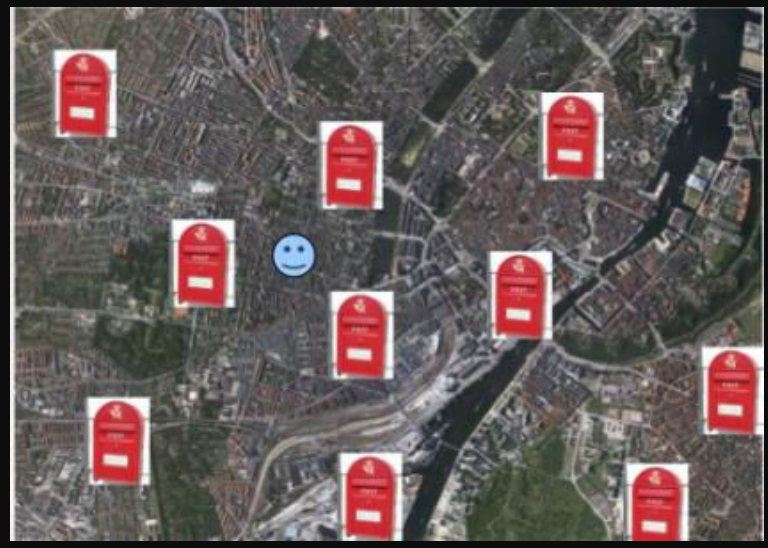
Voronoi Diagram

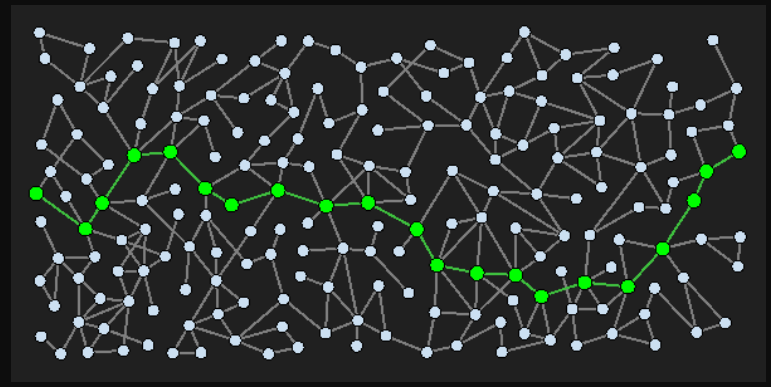


Voronoi Diagram



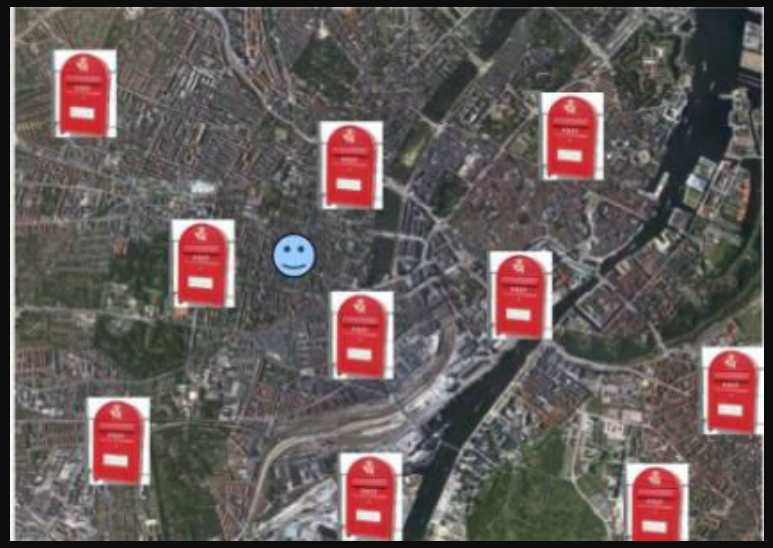
APPLICATIONS

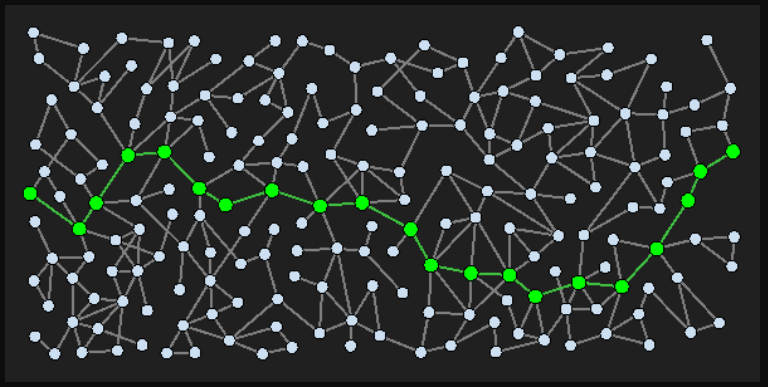




APPLICATIONS

- Biology (sinh học)
- Medical diagnosis (chẩn đoán y tế)
- Computer graphics (đồ họa máy tính)
- Ecology (sinh thái học)
- Computational chemistry (hóa học tính toán)
- Textures
- Zoology
- Image Compression
- v.v





Many thanks for your attention