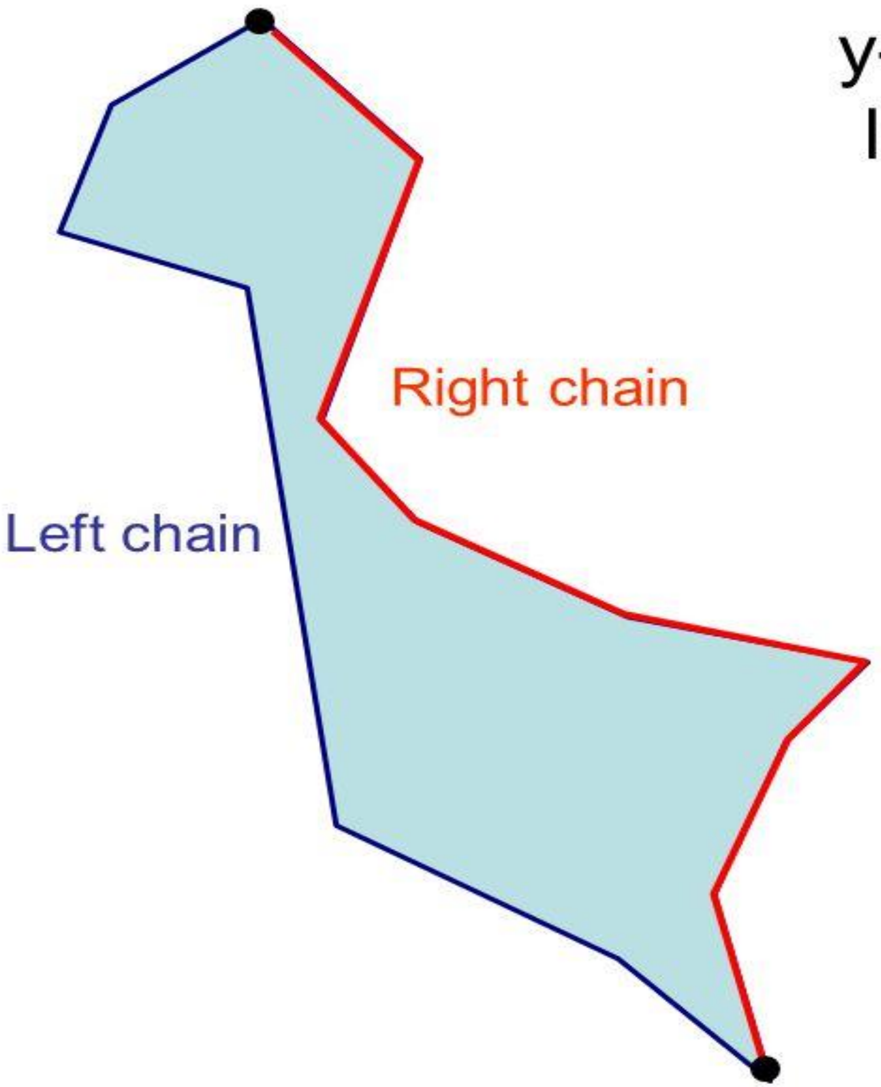
Polygon Partitioning

Partitioning the Polygon to Monotone polygons

A monotone polygon

- A polygon P is said to be monotone with respect to line L if ∂P can be split in to two polygonal chains such that each chain is monotone with respect to L
- The two chains share a vertex at either end

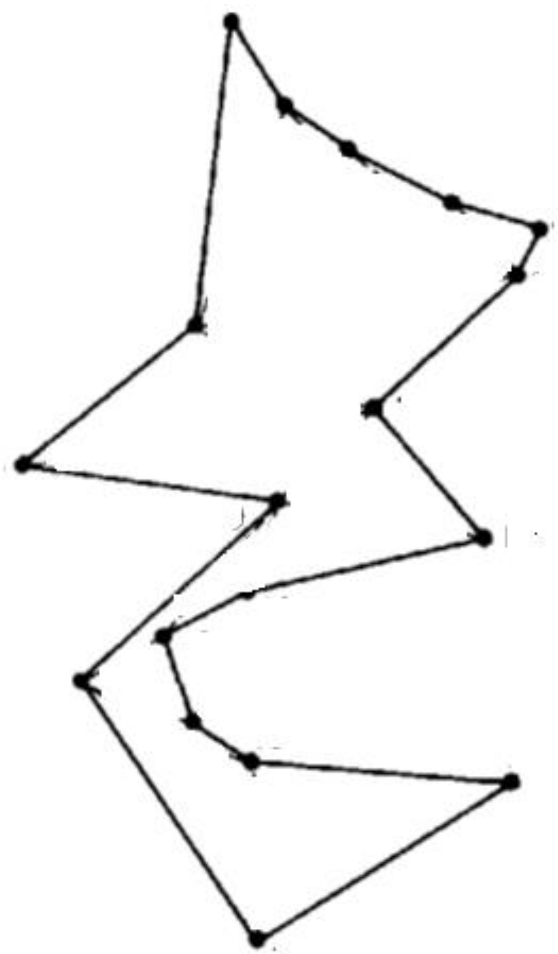
y-monotone polygon: left and right chains

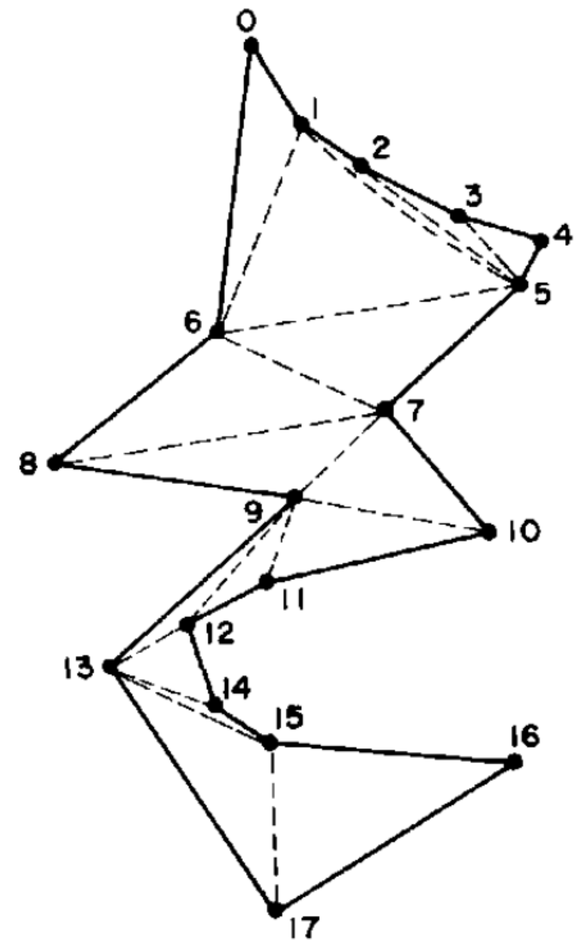


We will also assume that the polygon is strictly y-monotone, i.e. it is y-monotone and has no horizontal edges. Additionally, you may assume that no two vertices have the same y-coordinate

Triangulation of a monotone polygon

Input and Output





Time complexity

- Identify the major steps contributing to the time complexity

- ***Algorithm :* Triangulation of a Monotone Polygon**
- Sort vertices by decreasing y-coordinate, resulting in $p_0, \ldots, p_n$.
- Push p_0.
- Push p_1.
- **for** i = 2 **to** n - 1 **do**
- **if** p_i is adjacent to v_0 **then**
- **begin**
  - **while** $t > 0$ **do**
  - **begin**
  - Draw diagonal $p_i \longrightarrow v_t$.
  - Pop
  - **end**
  - Pop
  - Push v_t
  - Push p_i
- **end**
- **else if** p_i is adjacent to v_t **then**
- **begin**
  - **while** $t > 0$ and v_t is not reflex **do**
  - **begin**
  - Draw diagonal $p_i \longrightarrow v_{t-1}$
  - Pop
  - **end**
  - Push p_i
- **end**

Major steps :Time complexity

- Sorting in linear time
- Each vertex is pushed at most twice on the stack, once as *pi* and once as *vi.*
- Examination of the code shows that for each Push there is a corresponding Pop, and thus the algorithm requires *O(n)* time.

i	stack	condn	while	diag
2	0,1	else	No (1 refl)	
3	0,1,2	else	No(2 refl)	
4	0,1,2,3	else	No(3 refl)	
5	0,1,2,3, 4	else	Yes(4 notref)	5,3
5	0,1,2,3	Yes(angle 532 or angle 3 *not reflex*)		5,2
5	0,1,2	Yes(angle 521 or angle 2 *not reflex*)		5,1
5	0,1		No(1 ref)	
5	0,1,5 as pi			
6	0,1,5 as vt	if	Yes	6,5
6	0,1		Yes	6,1
6	0		No	
6	5,6			

Triangulation of a monotone polygon

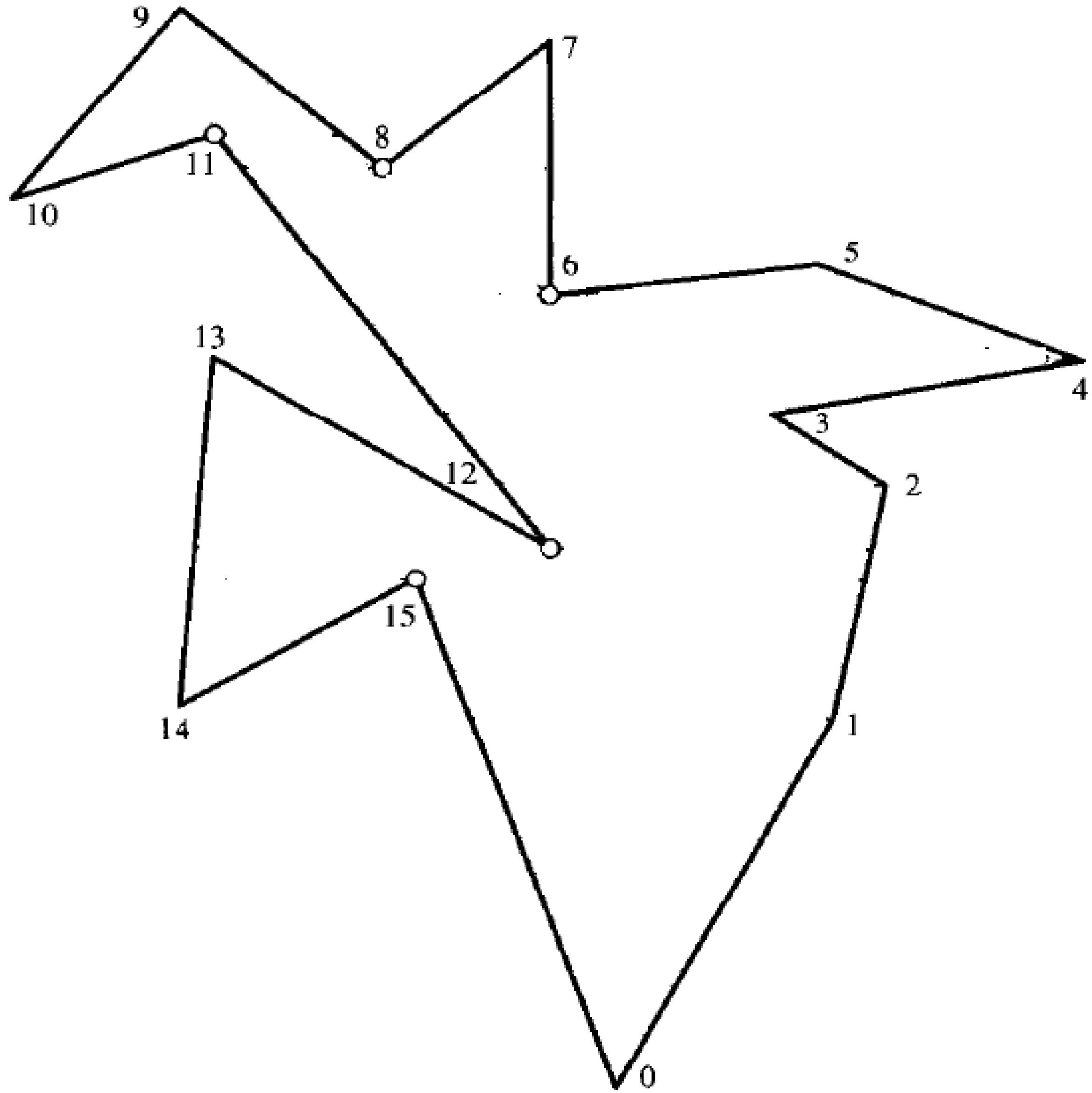
- **Triangulation of a monotone polygon can be done in linear time**

- Is it possible to use this algorithm (triangulation of a monotone polygon) to triangulate a non-monotone polygon (normal polygon) efficiently?
- The time complexity of the algorithm for triangulation of a normal polygon - $O(n^2)$

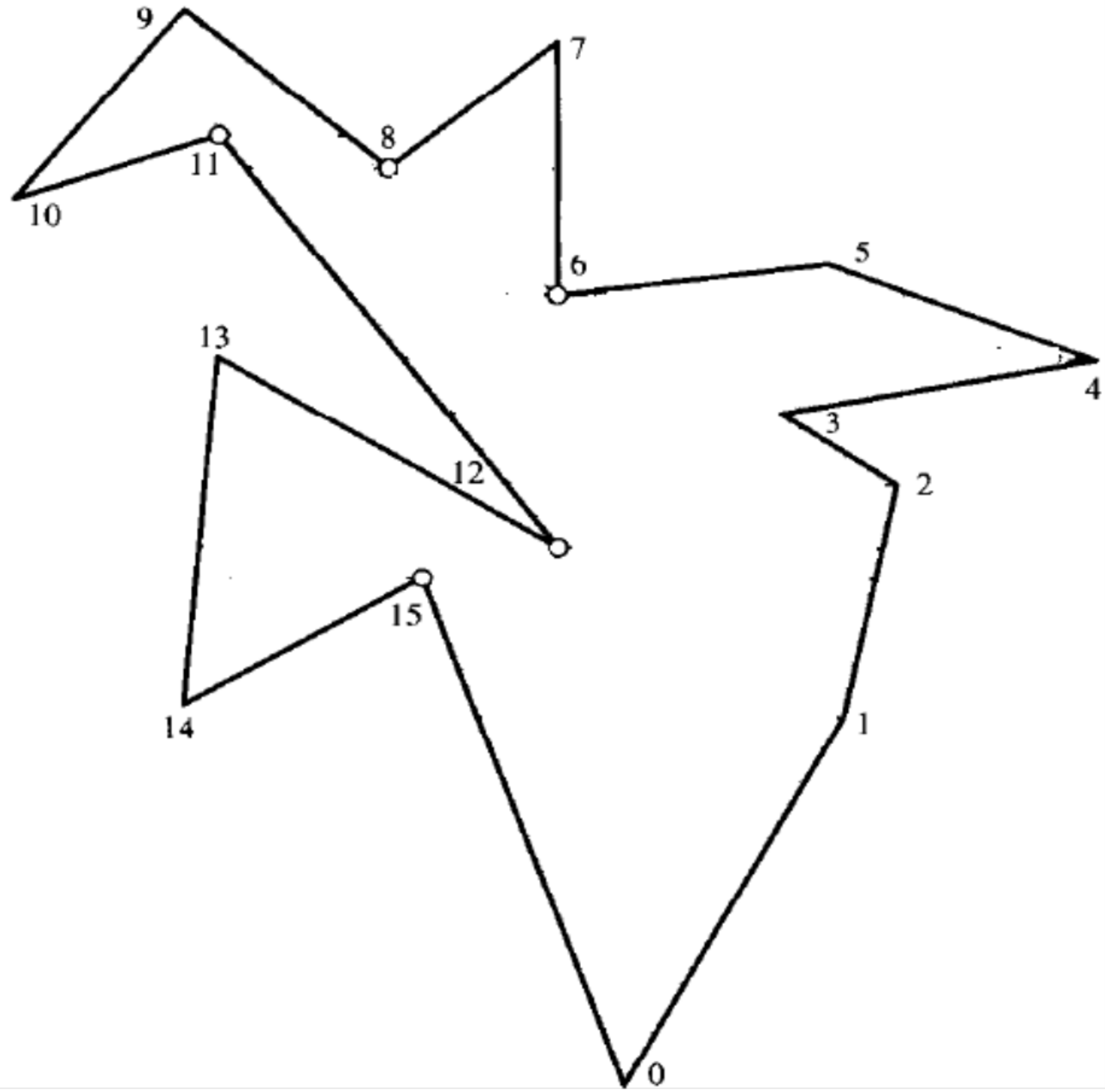
Algorithm to triangulate a non-monotone (normal) polygon

- Step 1: Partition a Polygon to monotone pieces
- Step 2 :Triangulate each monotone piece (can be done in linear time)
- If step 1 can be done efficiently (less than $O(n^2)$), then we can develop an efficient algorithm than the current $O(n^2)$ algorithm for triangulating a polygon
- We proceed focusing on a normal polygon

Is P monotone?

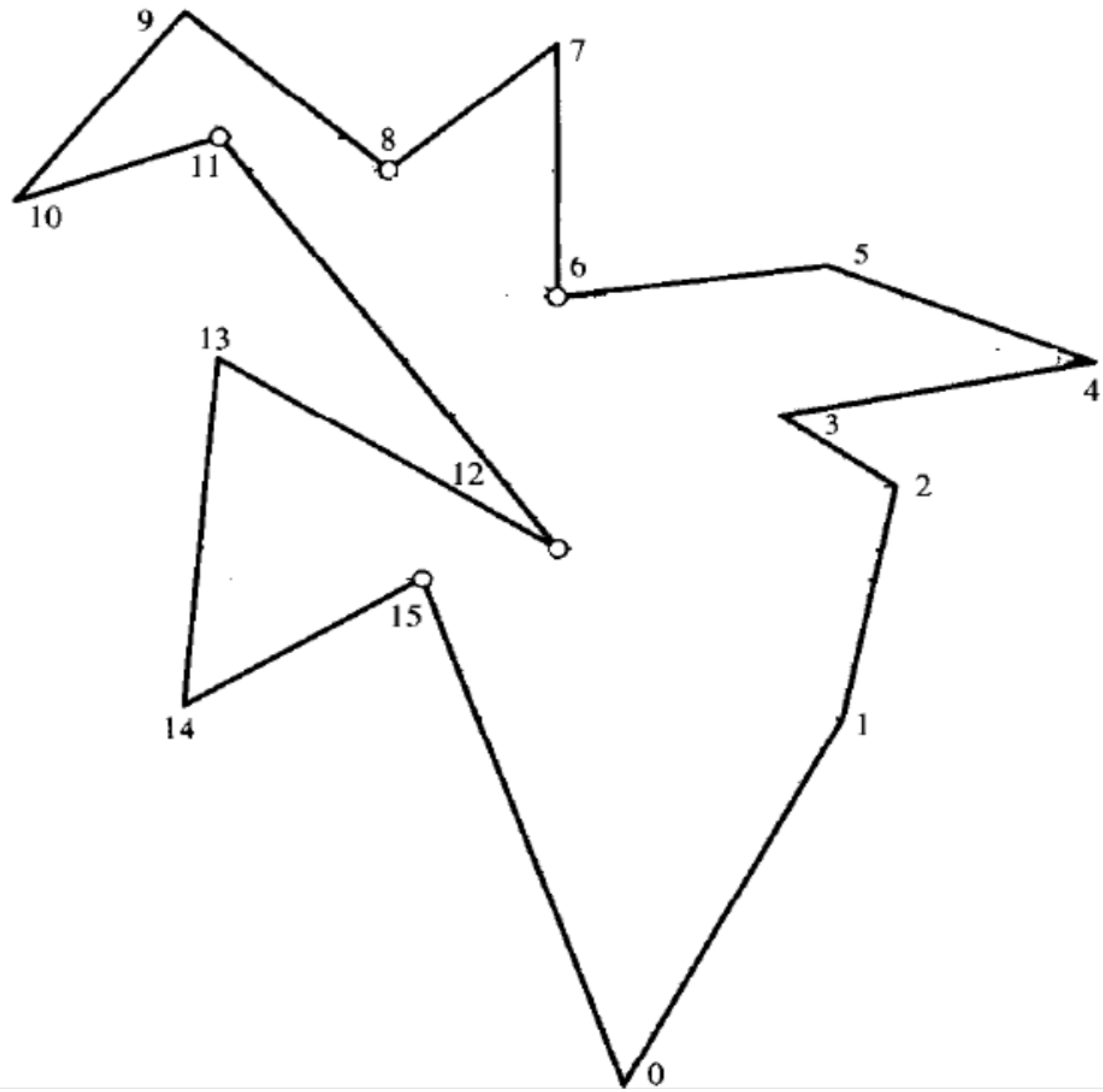


What characteristic makes P non-monotone?

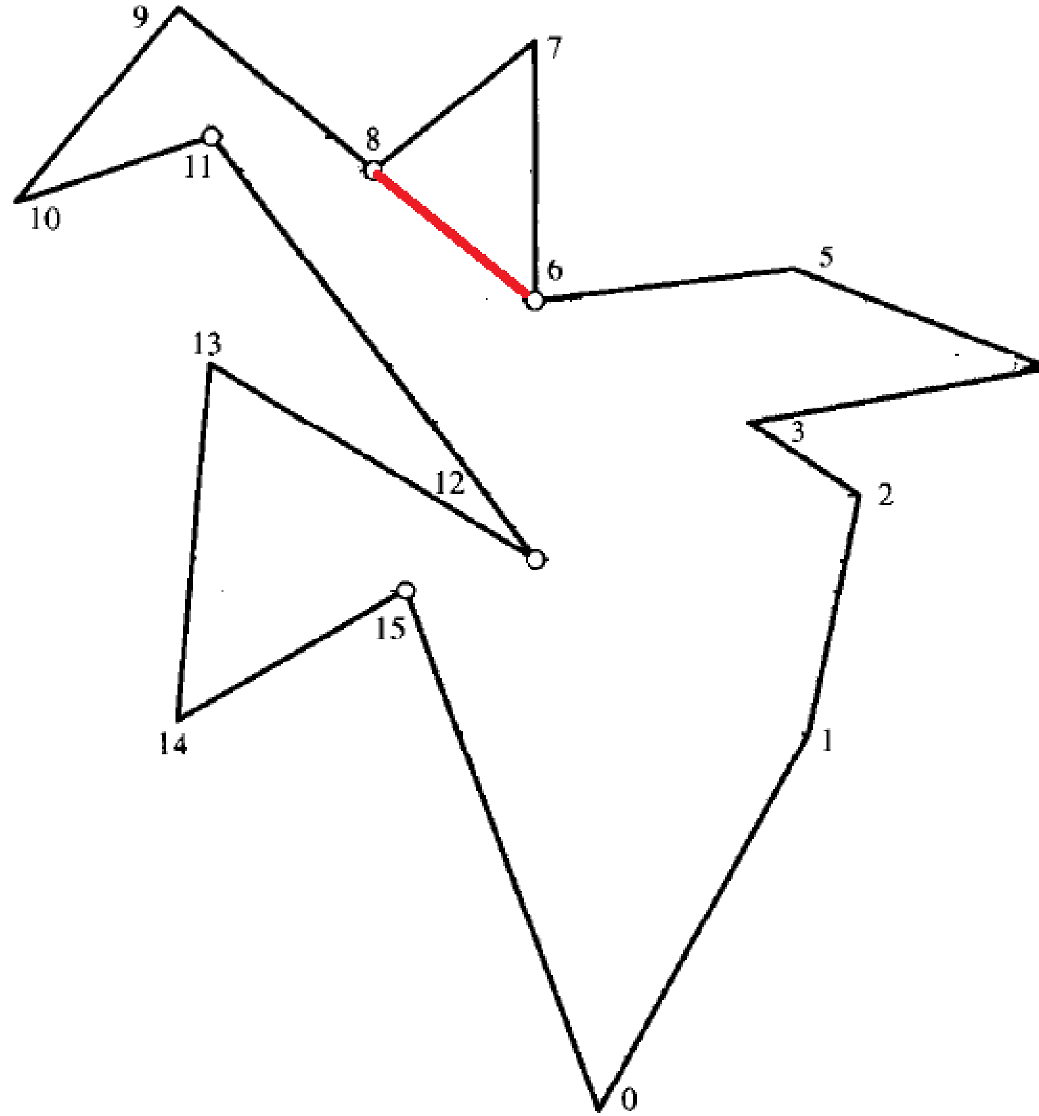


- Interior cusps

To make P monotone:



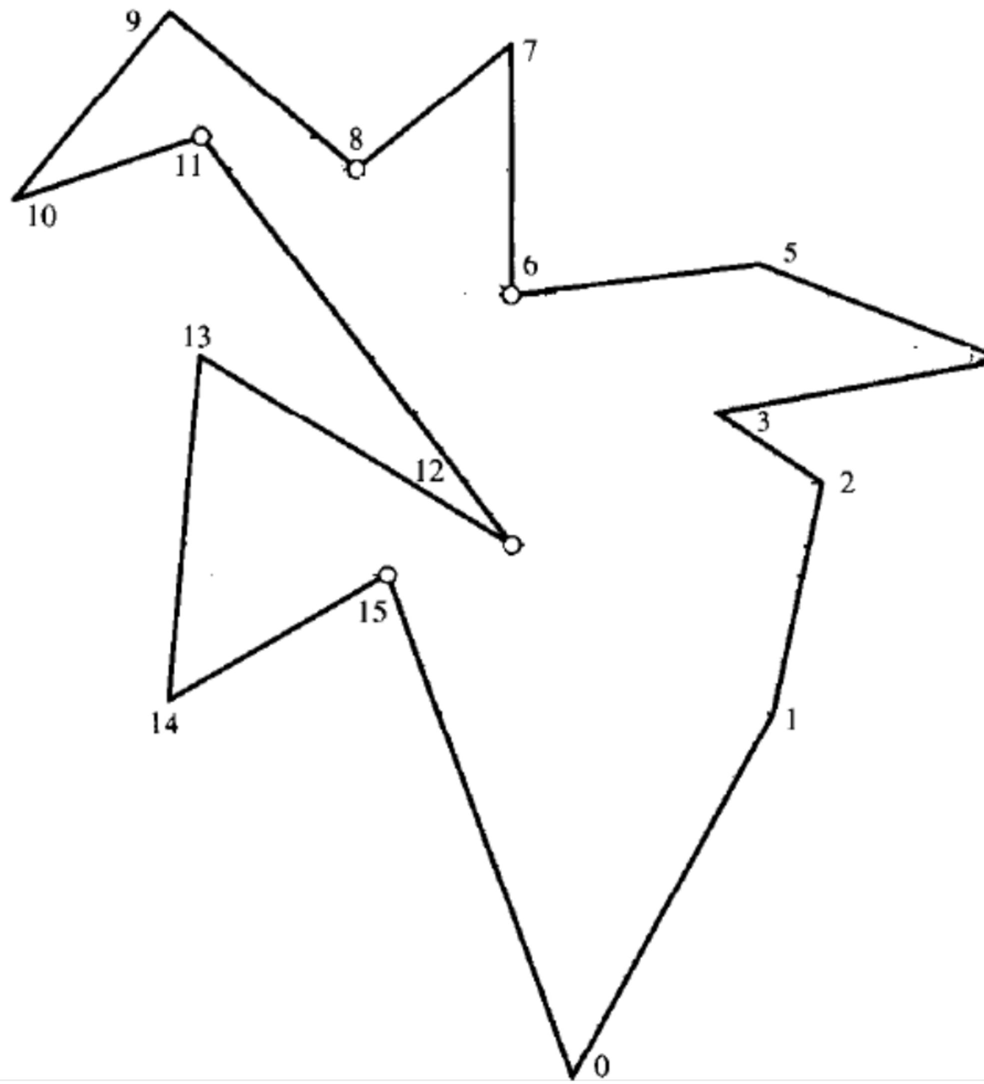
- Remove/ Break interior cusps
- Consider vertex 8



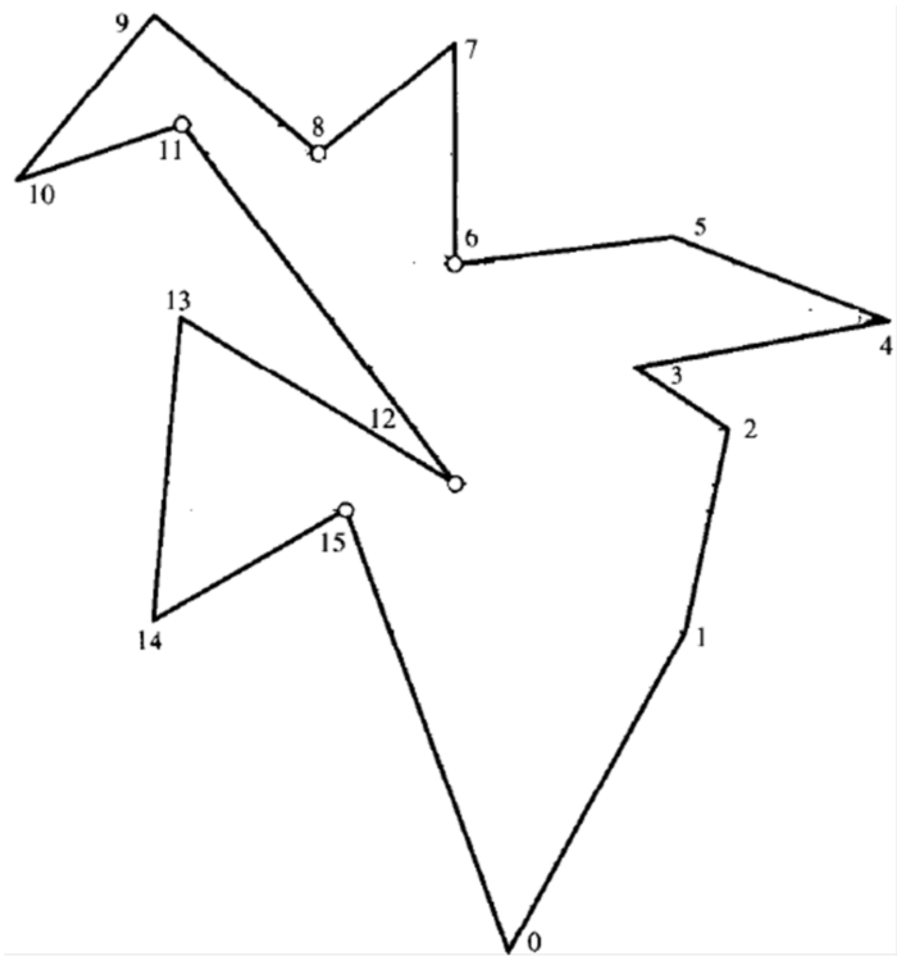
- Δ6 7 8 is monotone

To break non-monotonicity of vertex 8:

- We draw
  a line between 8 & 6

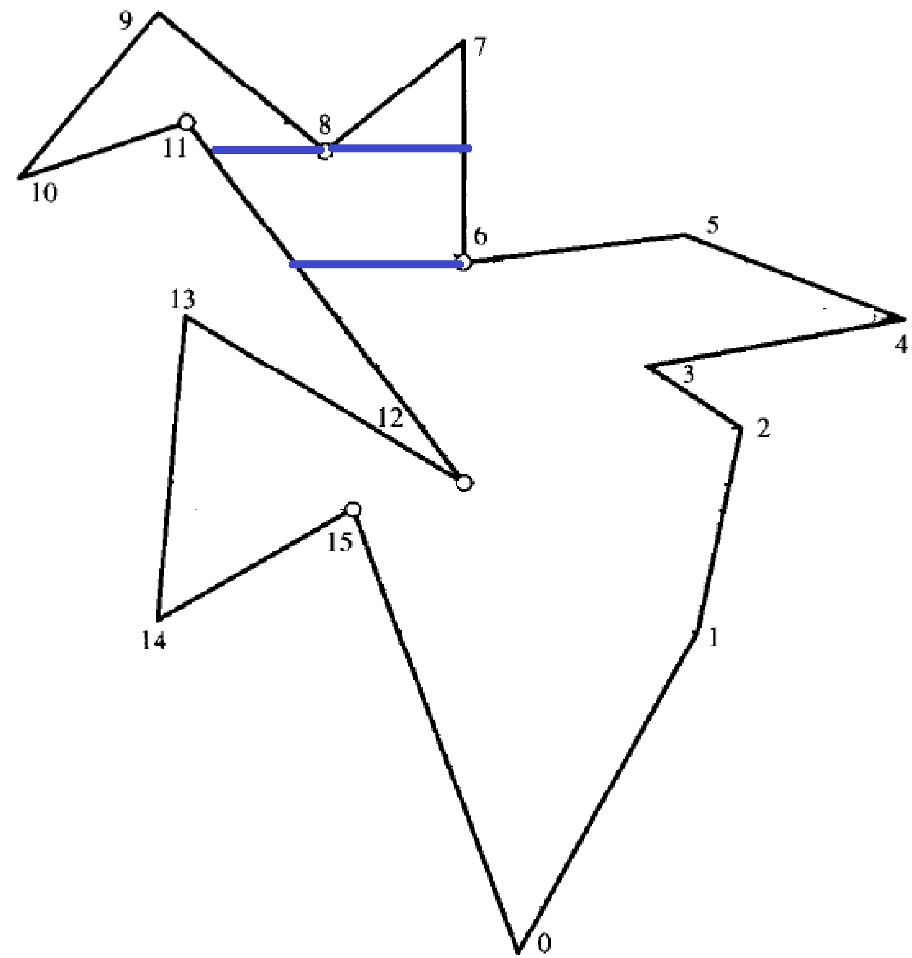


How do we select a vertex (eg: vertex 6) from many choices ?



Restrict our choice of a vertex to connect to :

- If we can restrict the choice of a vertex



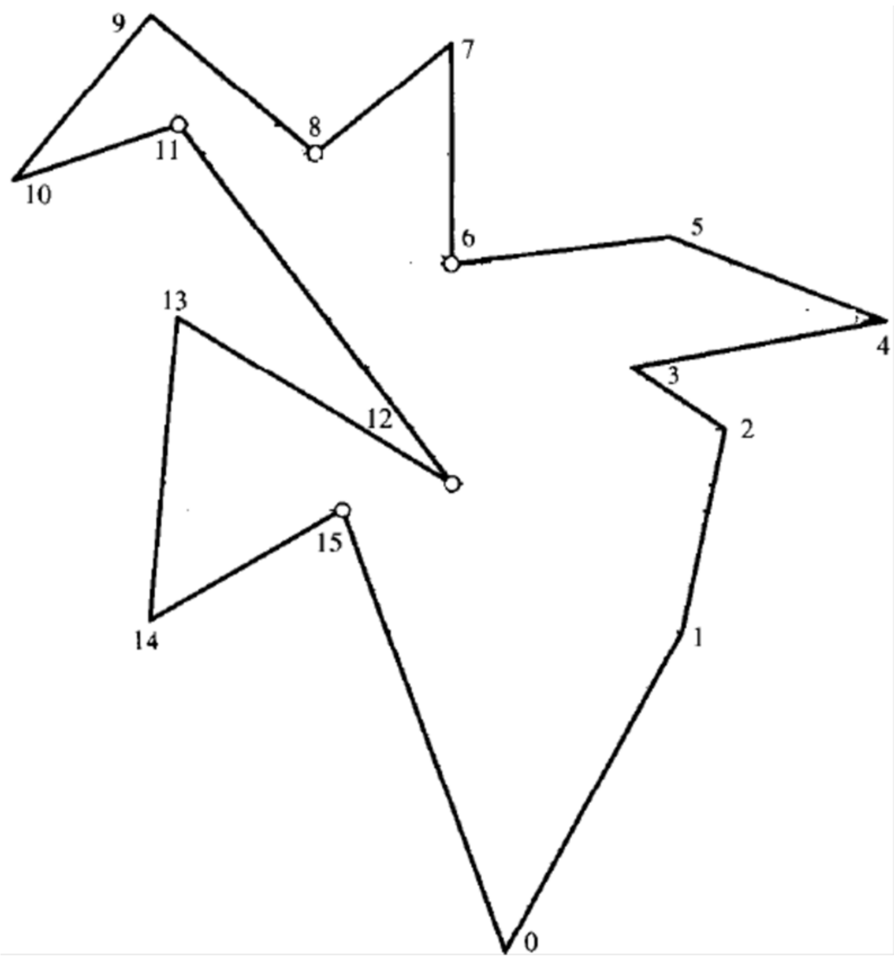
- Trapezoid
- What is a trapezoid?

Trapezoid

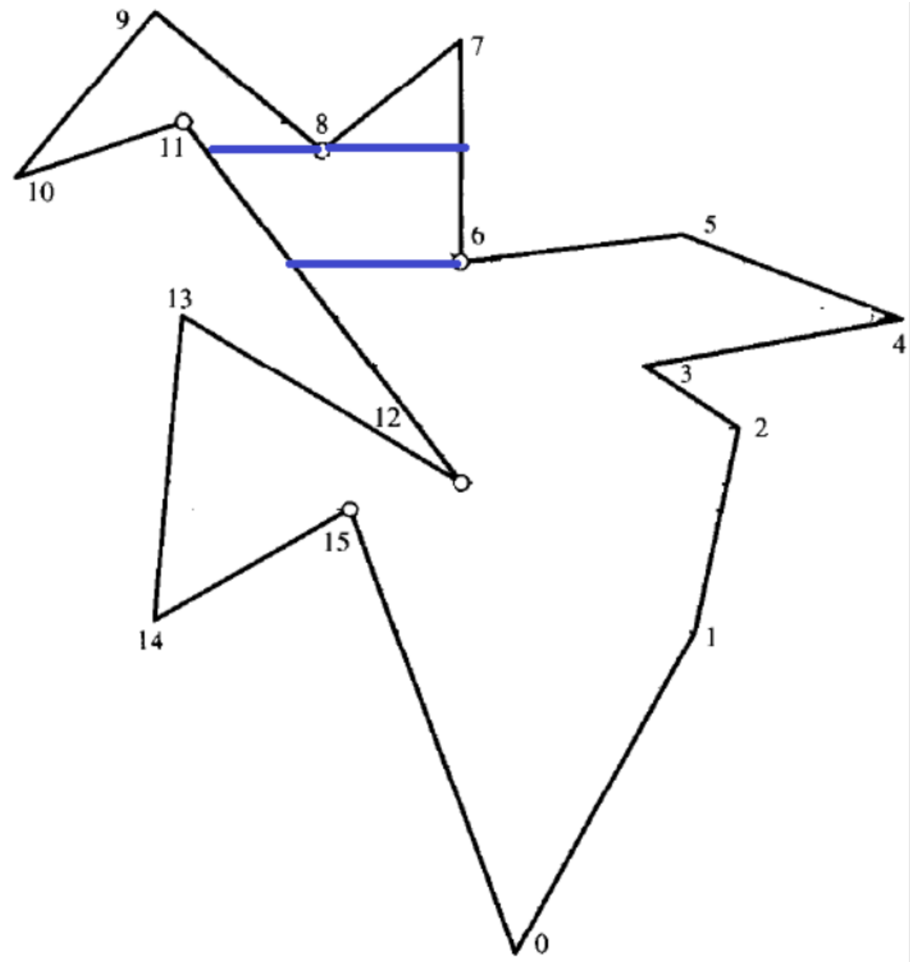
- Trapezoid: A convex quadrilateral with two parallel edges
- To summarize what we are doing up to now :
- We are trying to break the interior cusps of a Polygon
- By connecting vertex v_1 (reflex vertex which caused non-monotonicity) to another vertex v_2
- By introducing a trapezoid structure, we are restricting the choice for v_2
- First we trapezoidalize the polygon, then we divide it in to monotone pieces by breaking the interior cusps

Trapezoidalization

- How do we trapezoidalize P ?



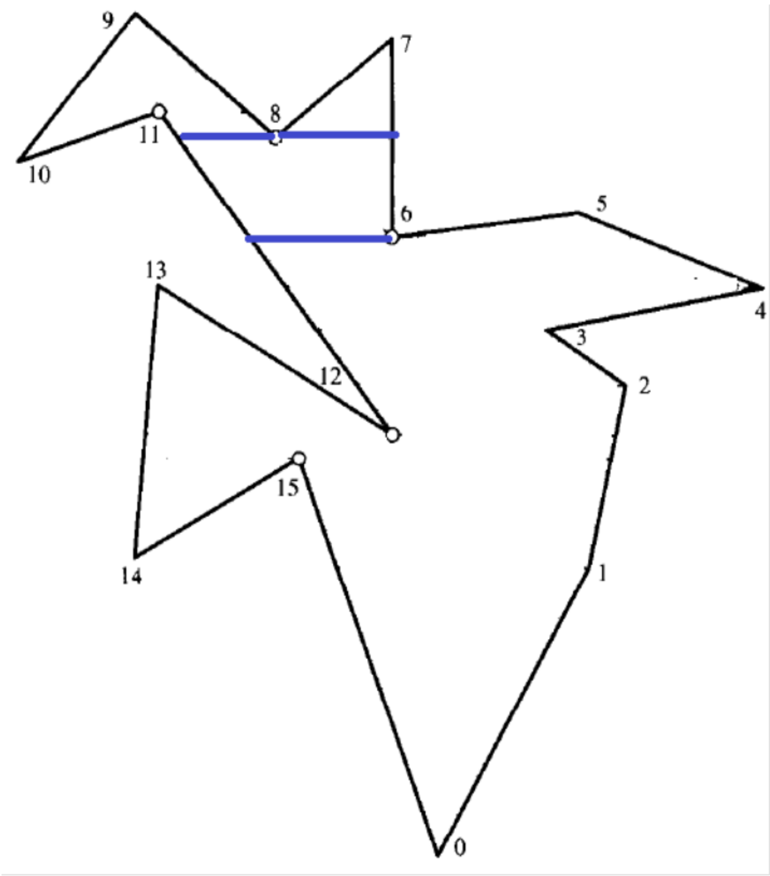
Trapezoidalization



- What have we done in this example?

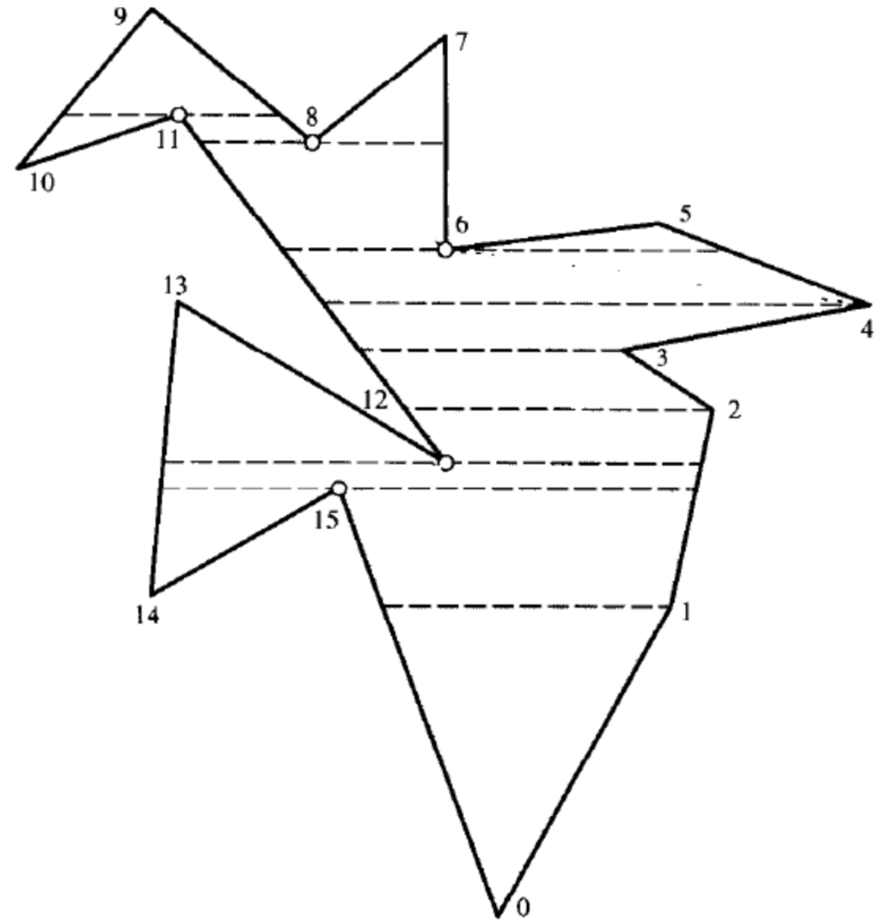
Trapezoidalization[Chazelle & Incerpi,1984]

- We have drawn horizontal lines through a vertex of a polygon
- We can either draw horizontal / vertical lines
- It can be a horizontal or vertical trapezoidalization.
- We focus on horizontal trapezoidalization.
- Assumption: No two vertices lie on a horizontal line



Horizontal trapezoidalization

- Polygon will be partitioned in to Trapezoids by drawing horizontal lines through a vertex of a polygon



- P completely trapezoidalized?

References

- J. O'Rourke: Art Gallery Theorems and Algorithms
- J. O Rourke, *Computational Geometry in C,* 2/e, Cambridge University Press, 1998 *)*
- [*https://www.cs.jhu.edu/~misha/Spring16/05.pdf*](https://www.cs.jhu.edu/~misha/Spring16/05.pdf)
  - *From John Hopkins University*

Thank you