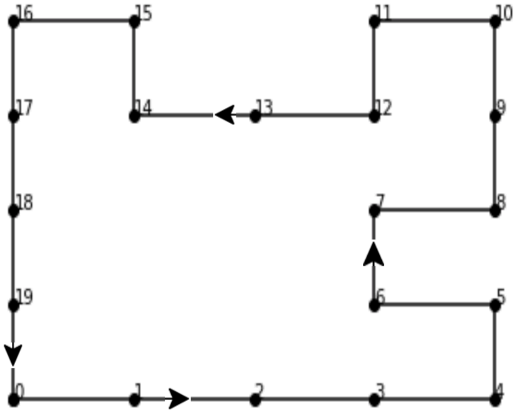
Partitioning of Rectilinear Polygons

Method of Labelling the graph

We take input as a rectillinear polygon from cursor keys, i.e., up(↑), left(←), and right (→). As input is read, the pointer proceeds forward and draws a rectillinear polygon with its trail. The labelling of the vertices starts from v_0 to v_{n-1}, and $v_0 = v_n$, where *n* is the *number of vertices in the polygon*.



A Rectillinear polygon consisting of 20 vertices.

Pressing a key once means going forward, left[1], or right[1]. A distance of only one unit can be traversed at a time.

INPUTS

G = Rectilinear Graph
X = Set of Abscissa of vertices
Y = Set of Ordinates of vertices
Collinear_Vertices = Set of Collinear Vertices[2]
Concave_Vertices = Set of Concave Vertices[3]
Horizontal_Chords = Set of Horizontal Chords[4]
Vertical_Chords = Set of Vertical Chords[4]

EXAMPLE

In the above figure, the pointer is shown by an arrow.

Total number of vertices = 20

Collinear_Vertices = [v_1, v_2, v_3, v_9, v_{13}, v_{17}, v_{18}, v_{19}]

Concave_Vertices = [v_6, v_7, v_{12}, v_{14}]

Algorithm for Finding Maximum partitions

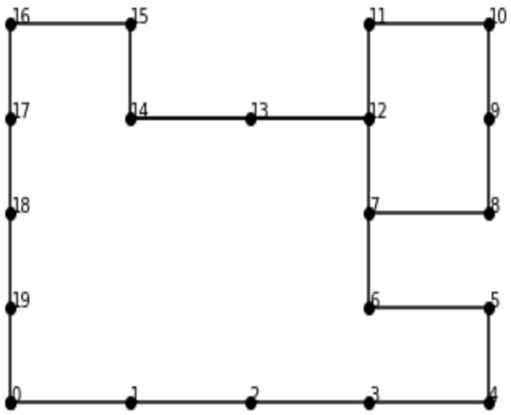
Maximum Partition: Partition of given rectillinear polygon into maximum number of non-overlapping rectangles.

STEP I

```
max_partition(G):
    for u in Concave_Vertices:
        for v in Concave_Vertices and v > u+1:
            if exists a chord joining v & u and ~exists another concave
             vertex on chord joining v & u:
                if chord is horizontal:
                    add (v, u) to Horizontal_Chords
                else if chord is vertical:
                    add (v, u) to Vertical_Chords
            else :
                loop_back
```

Task Achieved: All the edges that exist between *any two concave vertices* are being added to their *respectful categories*.

Example:



Horizontal_Chords = ∅

Vertical_Chords = [(v_7, v_{12})]

Explanation:

u > v : Comparison between two vertices is done on the basis of their respective vertex indices[5].

Here ***v-u*** should be greater than unity, because this assures the vertex v is not consecutive to u and has a higher index than u. Thus, iteration through each pair of vertex is done only once, making it more efficient.

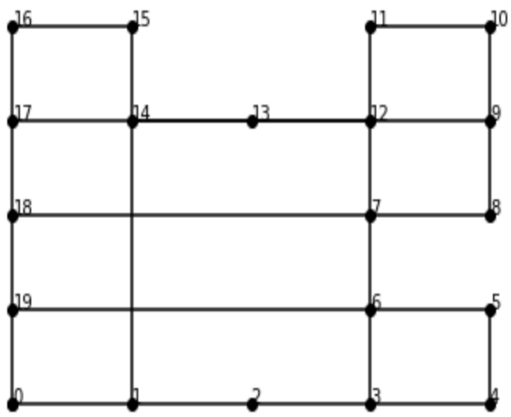
In the above code, we iterate through all (concave vertex, concave vertex') pairs, and check for existence of vertical and horizontal chords, that are not intersected by any other vertex. We observe that, v_7 and v_{12} are the only two concave vertices and between whom, there exists a vertical chord. Therefore, it is added to the set of *Vertical_Chords*. Also, there does not exist any horizontal chord between any two concave vertices and therefore, set of *Horizontal_Chords* is empty.

STEP II

```
for u in Collinear_Vertices:
    for v in Concave_Vertices:
        if exists a chord joining v & u and ~exists another concave
            or collinear vertex on chord joining v & u:
            if chord is horizontal:
                add (v, u) to Horizontal_Chords
            else if chord is vertical:
                add (v, u) to Vertical_Chords
        else :
            loop_back
```

Task Achieved: All the chords between *collinear vertices and concave vertices* are being added to their *respective categories*.

Example:



Horizontal_Chords = [(v_9, v_{12}), (v_{17}, v_{14}), (v_{18}, v_7), (v_{19}, v_6)]

Vertical_Chords = [(v_7, v_{12}), (v_1, v_4), (v_3, v_6)]

Explanation:
In the above code, we iterate through all (collinear vertex, concave vertex) pairs, and check for existence of vertical and horizontal chords between them, that are not intersected by any other vertex.
If any chord is found, it is added to set of *Vertical_Chords* or *Horizontal_Chords*, depending on its orientation.

STEP III

Thus, we have found all the chords, and only need to plot them now.

```
    plot(X,Y)
    plot(Horizontal_Chords)
    plot(Vertical_Chords)
    display(plot)
```

1. Left and Right operations changes the direction the pointer faces. ↩

2. Vertices that are induced after going forward consecutively. Although in the example, they are not explicitly shown, but they do exist and at a distance of one unit from its previous vertex. ↩

3. If the interior angle made by the two edges incident at this vertex is 270 degree. ↩

4. Chords are lines joining two vertices which are not already part of the polygon. ↩

5. As, the way of labelling is defined, there is unique labelling of each rectillinear polygon. ↩