TOPIC IN ALGORITHMS

Final Report

Art Gallery Problem Implementation

(Vertex guard)

Prepared by :
TGDK SUMANATHILAKA
B150413CS
sumanathilaka_b150413cs@gmail.com

Instructor : DR.SUBHASREE

COURSE : CS4027
DATE : 05/ 04 /2018

TABLE OF CONTENTS

Problem Specification

The **art gallery problem** or **museum problem** is a well-studied visibility problem in computational geometry .It originates from a real-world problem of guarding an art Gallery with the minimum number of guards who together can observe the whole gallery.

Problem :

Given a 2D Graph ,What is the minimum number of guards to safeguard the painting of the art gallery .(Guards can only placed in vertices , guards have 360 degree visibility)

INPUT OUTPUT Specification

Input

1) The no of vertices of the art Gallery.
2) (X Y) plot of each vertex.

Eg: (1)	(2)	(3)
4	5	6
1 1	0 0	0 0
1 0	0 2	2 4
0 1	1 0	3 2
0 0	1 3	5 5
	2 1	3 4
		1 4

Output

1) Minimum No of guards .
2) (X Y) location of guards.

Eg : (1)	(2)	(3)
No of guards : 1	No of guards : 1	No of guards : 2
Location :(0 0)	Location : (0 0)	Location : (0 0) (3 4)

Discussions On Vertex guard algorithm:

Method 1 :

Step 1 :Draw lines through every pair of vertices of **P** and compute all convex components **C1,C2,C3,.....Cm** of **P** . Let **C=(C1,C2,C3,.....Cm), N=(1,2,....,n)** and **Q = Ø.**

Step 2 :*For* **i <= j<= n,** construct the set **Fj** by adding those convex components of P that are totally visible from the vertex **Vj.**

Step 3 :Find **i∈N** such that **|Fi| >= |Fj |** *for all* and **j ∈ N and i** $\neq$ **j.**

Step 4 :Add **i** to **Q** to and delete **i** from **N .**

Step 5 :*For all* ***j*∈ Ni.Fj = Fj-Fi** , and **C = C- Fi.**

Step 6 :*If* **|*C*|** $\neq \emptyset$ *then goto* Step 3.

Step 7 :Output the set Q and Stop.

Time Complexity of the Algorithm

Step 1 :
Since O(n^2) lines are drawn in P to compute components , m can be at most O(n^4).
Step 2 : 0(mn)
Step 3 : O(n^2)
Step 4 : O(1)
Step 5 : O(n^5)
Hence the overall time complexity of the approximation algorithm is **O(n^5)**

Method 2 :

Step 1 :

Triangulate the given polygon **P** by adding non overlapping diagonals.

a) Using Two ear theorem ***
b) Recursive
c) Monotone polygon principles

Step 2 :

3 colour the triangulated polygon using 3 colour principles.

Step 3:

A Single Color will cover the whole polygon (Take the min no of vertices with same color)

Step 4 :

Pigeonhole principle

Time Complexity of the Algorithm

Step 1 :

a) Using Two ear theorem **O(n^3)**
b) Recursive **O(n^2)**
c) Monotone polygon principles **O(nlogn)**

Step2 :

3 colouring of graph : **O(1.3286^n)**
K colouring : **O(2^n .n)**

Results of the graphs with the implementation.

Implementation was done in C
The Algorithm is ;

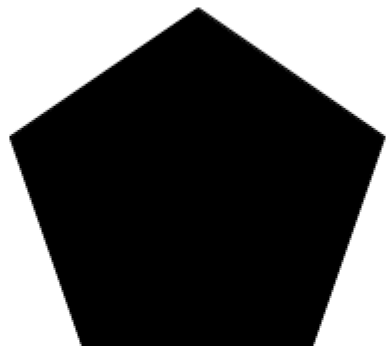
1. Insert the Point in counter clockwise
2. Triangulate the Polygon (using two ear theorem)
3. 3- colouring theorem
4. Min no of guards (least coloured vertexes)

Test Cases

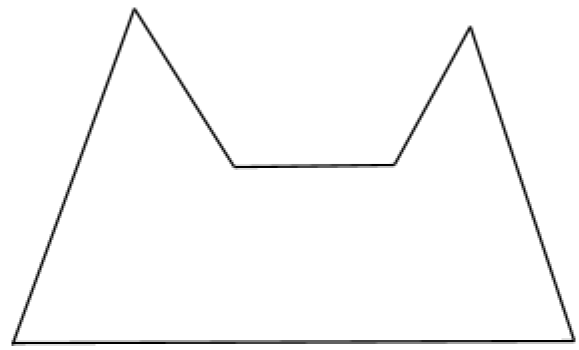
NO	NO of vertices	Test cases	Min no of guard	points
1	4	(0 0) ,(1 0),(1 1), (0 1)	1	3 (0 1)
2	5	(0 0),(2 0),(3 1),(2 2),(0 1)	1	4 (0,1)
3	6	(0 0),(10 0),(8 4),(6 2),(4 2),(2 4)	2	2 (8 4) , 4 (4 2)
4	9	(0 0) ,(14,0),(14,10),(12,5),(10,10),(8,5),(4,10),(2,5),(0,10)	2	3 (12,5) , 7 (2,5)
5	9	(0,0),(16,0),(14,10),(12,5),(10,5),(8,10),(6, 5),(4,5),(2,10)	3	2 (14,10), 4(10,5) , 7(4,5)
6	12	(0 0) ,(6 2) ,(12 0),(8 4),(10 6),(12 6),(8 8),(6 6),(2 8),(0 6),(2 6),(4 4)	3	6(8 ,8) , 9 (0,6) 11(4 ,4)
7	12	(0,8),(4,8),(2,2),(8,9),(6,0),(12,4),(10,4),(10,12),(6,9),(6,14),(2,11),(4,11)	4	2(2,2) , 4(6,0) , 8(6,9) ,11(4,11)

Shape 1

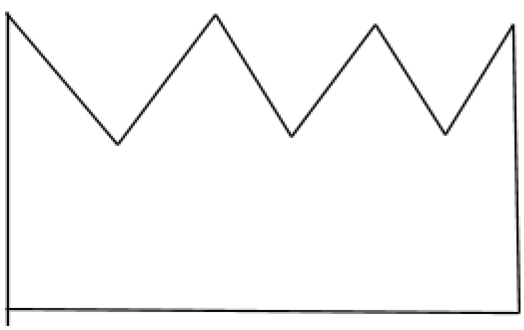
shape 2



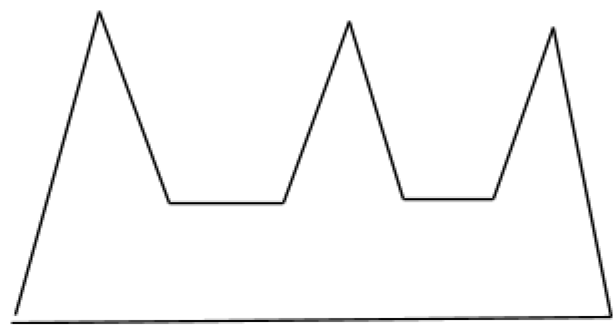
shape 3



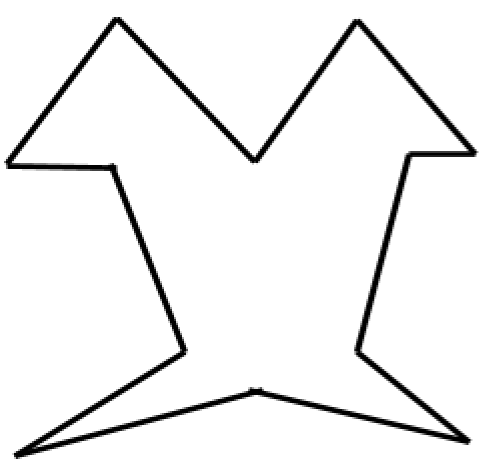
Shape 4



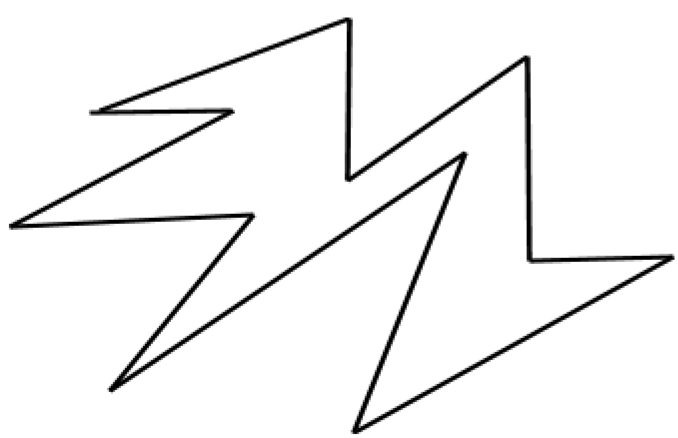
shape 5



shape 6



shape 7



Reference

1. Computational Geometry in c - by Joseph o'Rourke
2. https://en.wikipedia.org/wiki/Art_gallery_problem
3. https://www.cs.purdue.edu/homes/aliaga/cs635-10/lec-artgallery.pdf
4. http://www.math.iit.edu/~kaul/talks/LongArtGalleryTalk.pdf
5. http://www.geeksforgeeks.org/backttracking-set-5-m-coloring-problem/