Amirkabir University of Technology
(Tehran Polytechnic)

# Report

# Star-Shaped polygon of N points with an arbitrary $S_0$

Professor:
Dr. Mohades (mohades@aut.ac.ir)

Student:
Mohammadreza Ardestani (**9513004**)
9, Dec, 2020

## Table of content:

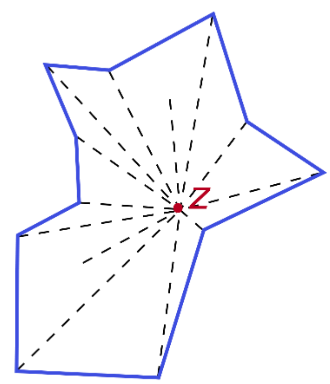


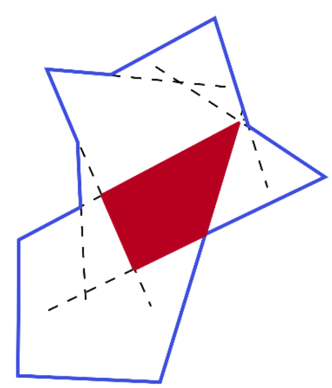
## Introduction and definitions

**Definition of star shaped polygon:**

A star-shaped polygon is a polygonal region in the plane that is a star domain, that is, a polygon that contains a point from which the entire polygon boundary is visible. In other word, star shaped polygon is a polygon that its kernel of visibility is not null.
(for more information check: *Computational Geometry and computer graphics in C++)*

**Example of star shaped polygon:**

## Algorithm

**Description of the problem:**

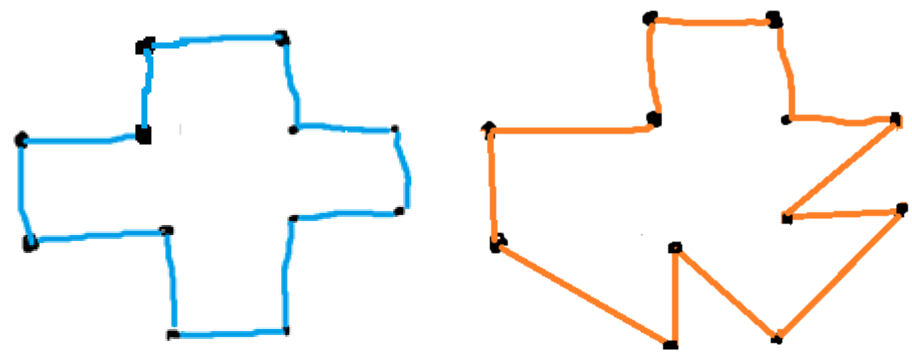
We have N different number of points (whether online or offline) and then we return a polygon chain such that its kernel is not null (and based on our algorithms one of those N points are element of the Kernel and we call that point $S_0$).

(I will leave the interested reader (perhaps you!) with this question:

> can we always convert any star shaped polygon into star shaped polygon such that at least one of its vertices ($S_0$) is an element of the kernel of visibility? )

Star shaped polygon always exits (due to the following algorithms) and it's not unique.

**2 different star shaped polygon with the same set of points:**



**Variation of this problem:**

1. Off line given points and an arbitrary $S_0$
2. Off line given points and lowest point as $S_0$
3. Online given points and first points as $S_0$

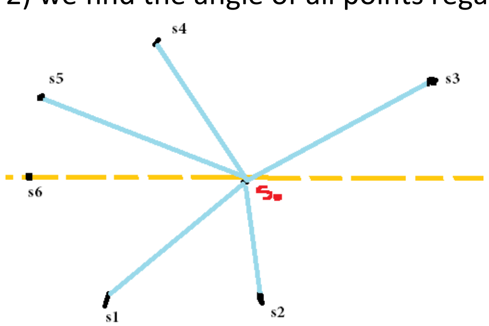
Variation 2,3 are reducible to number 1.

So we will solve the first one and then we can use it for solving 2,3 as well.

**Diving into the algorithm:**

1) At first step we generate N random points and one of them as $S_0$.

2) we find the angle of all points regarding to the Y = $y_{s_0}$ :



For instance in this case angles of vertices would be :
180 for S6 and 160 for S5 and 100 for S4 and 40 for S3 and - 80 for S2 and – 150 for S1.

3) then we sort all points by their angles.

4) by $S_0$ and all points belong Y = $y_{S_0}$ we build a star shaped polygon.

by $S_0$ and all points belong Y = $y_{S_0}$ we build a star shaped polygon.

we do it in this way :

We connect $S_0$ into the first lowest angle and then we connect I'th angle into (i+1)'th angle and finally we connect the last one into $S_0$.

( Question: Prove these two polygons are star shaped and $S_0$ is an element of the Kernel.)

( note: if we don't have any point above and belong the Y = $y_{S_0}$ the probem is solved and we don't execute part 5.)

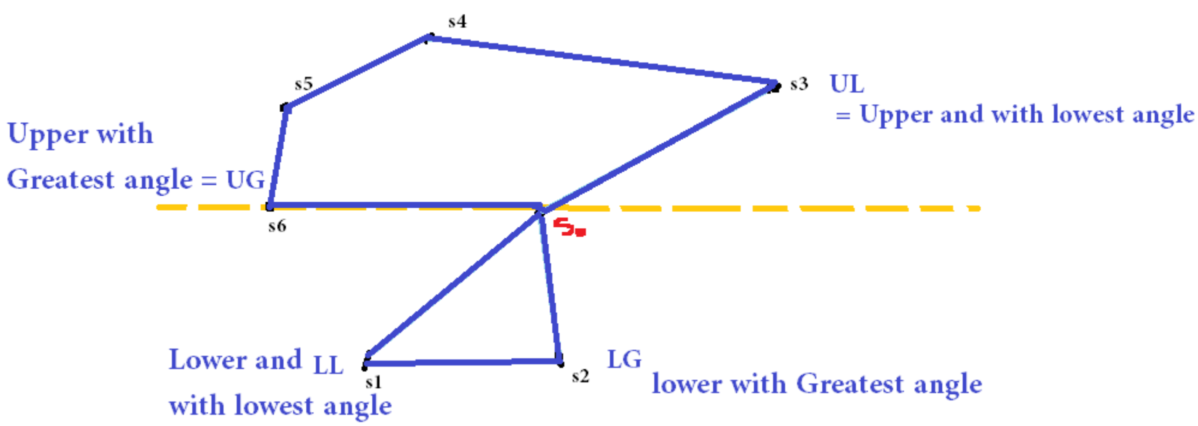
( for instance at the end of this step for the figure in part 3 we have 2 star shaped polygon :

$S_0\ S_3\ S_4\ S_5\ S_6\ S_0 \quad and \quad S_0\ S_1\ S_2\ S_0$ )

5) Merging two star shaped polygon:

Now that we have come up with 2 star shaped polygon we need to merge them.

For increasing clarity I introduce 4 type of points: UL, UG, LG , LL.




(if we only have one point belong (or above) the line LL and LG are equal.)

For merging these two Star shaped polygon we do as follow:

If ( UL $S_0$ LG is not reflex angle ):

merge these 2 polygon by removing LG $S_0$ edge and UL $S_0$ edge and add an edge UL LG

else:

merge these 2 polygon by removing LL $S_0$ edge and UG $S_0$ edge and add an edge UG LL

(Legitimize step5 by proving final polygon remains starShaped and$S_0$is an element of theKernel)

**Time complexity of algorithm:**

1) O(n)
2) O(n)
3) O (n log n)  // using Merge sort
4) O (n)
5) O(1)

T(n) = O( n log n)

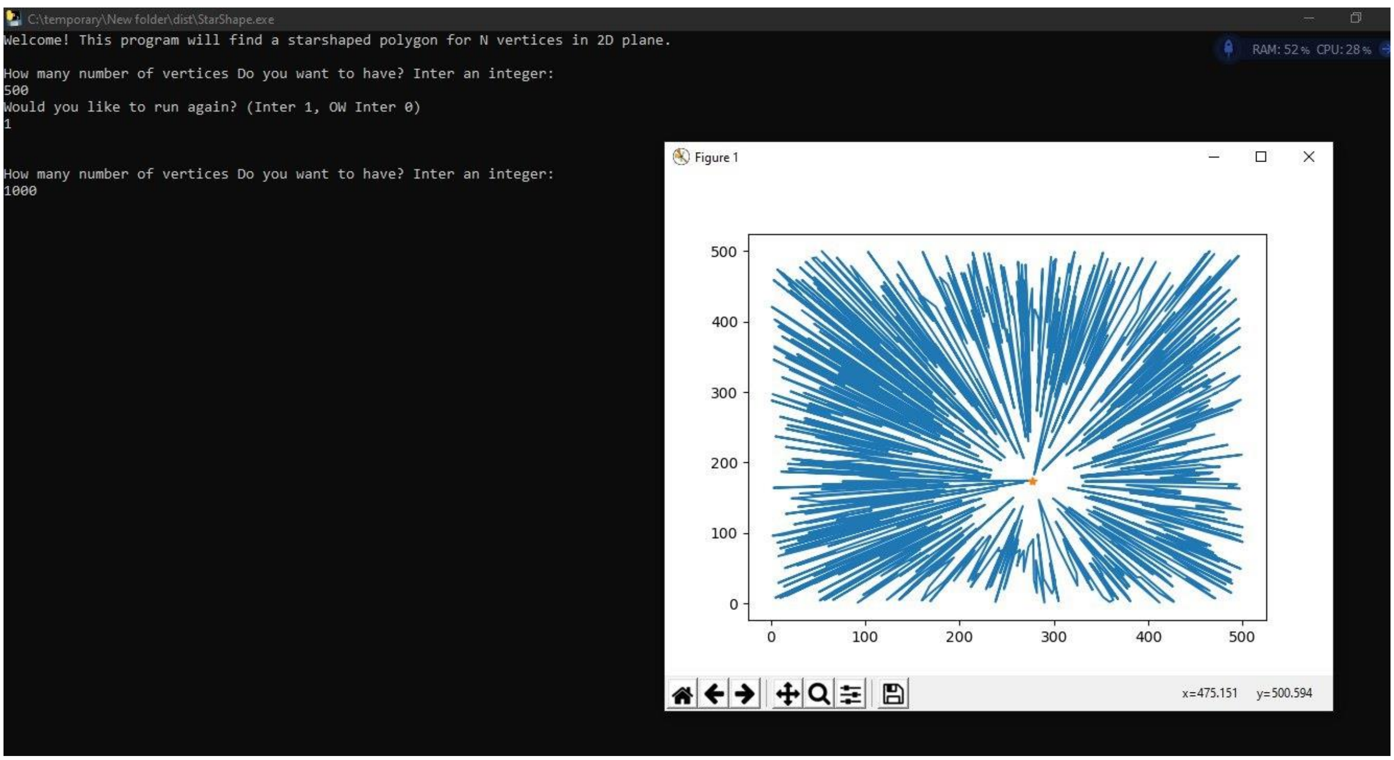
## Implementation with Python

**Here is the link for .py and .exe file:**

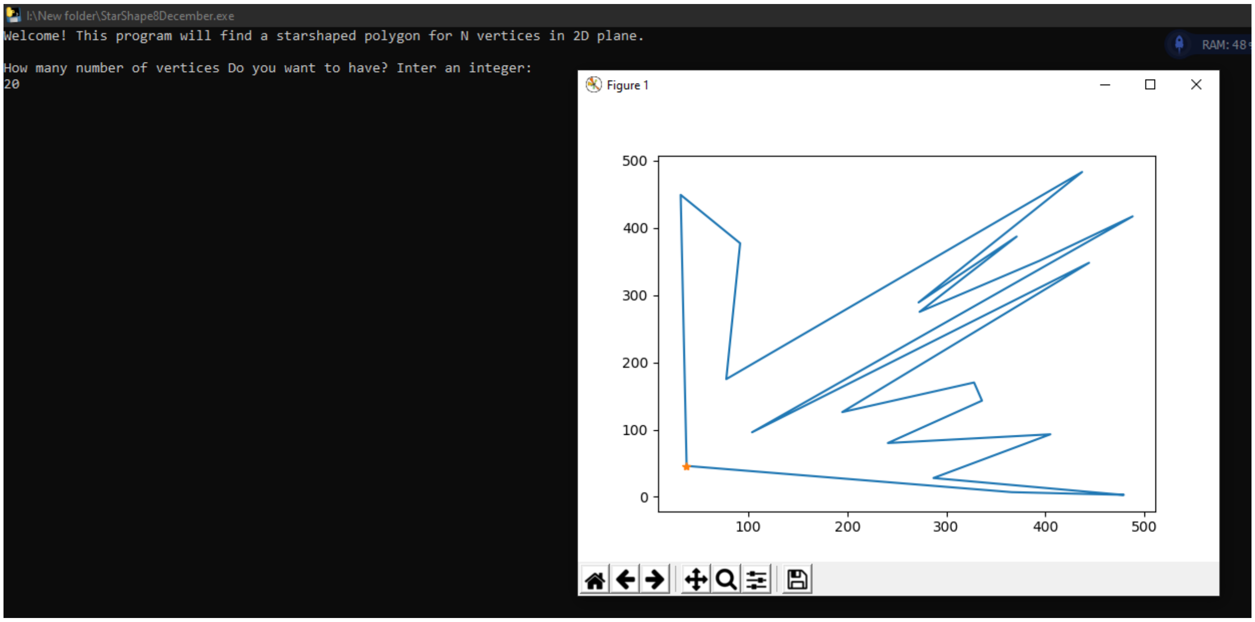
**.py [link](link)        .[exe link](exe link)**

**Sample output:**

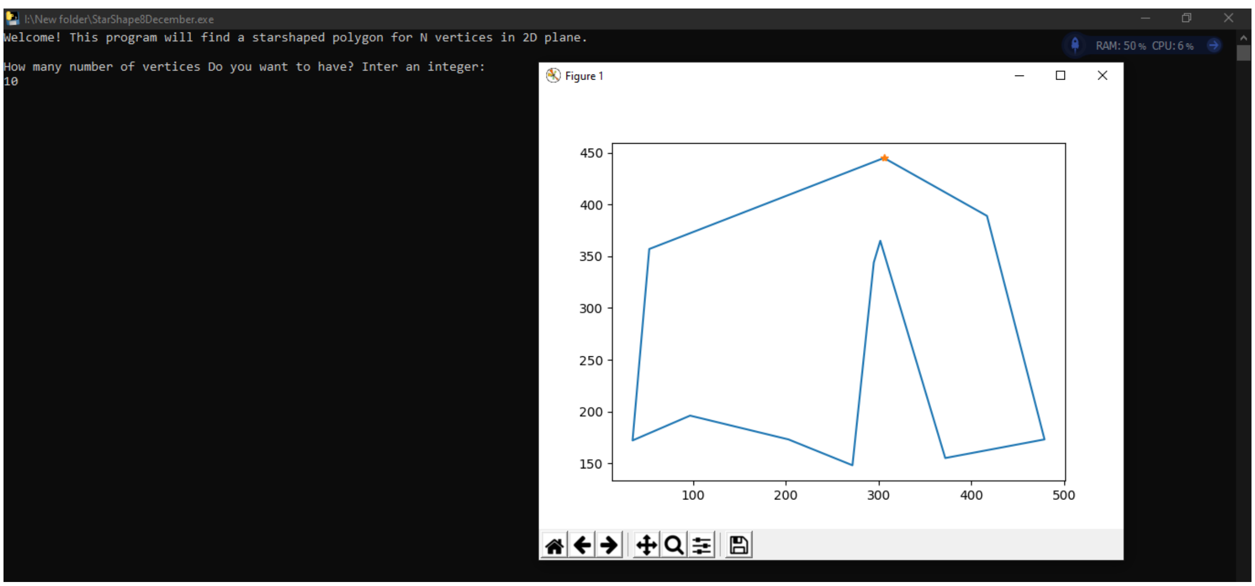
**N = 1000**



N = 20




N = 10




**Thanks for your attention.**