

**Instrucciones:**

- ◇ Fecha de publicación: 23 de febrero de 2021 a las 9:00 am.
- ◇ Fecha de entrega: 25 de febrero de 2021 hasta las 11:59 pm.
- ◇ Medio de entrega: <https://e-aulas.urosario.edu.co> (no se reciben entregas por correo electrónico u otros medios).
- ◇ La actividad **debe** realizarse **individualmente**.
- ◇ **Formato de entrega:** Archivo PDF conteniendo la solución al problema del enunciado.

**Protocolo para la evaluación:**

Los siguientes lineamientos serán seguidos de forma estricta y sin excepción.

1. Solamente será posible tener acceso a [e-aulas.urosario.edu.co](https://e-aulas.urosario.edu.co) y a los sitios web correspondientes a la documentación de Python y C++ dispuestos por el profesor.
2. Maletas, morrales, bolsos, etc. deben estar ubicados al frente del salón.
3. Celulares y otros dispositivos electrónicos deben estar apagados y ser guardados dentro de las maletas antes de ser ubicadas en su respectiva posición.
4. El estudiante no debe intentar ocultar ningún código que no sea propio en la solución a la actividad.
5. El estudiante solo podrá disponer de hojas en blanco como borrador de apuntes (opcional).
6. El estudiante puede tener una hoja manuscrita de resumen (opcional). Esta hoja debe estar marcada con nombre completo.
7. E-aulas se cerrará a la hora en punto acordada para el final de la evaluación. La solución de la actividad debe ser subida antes de esta hora. El material entregado a través de e-aulas será calificado tal como está. Si ningún tipo de material es entregado por este medio, la nota de la evaluación será 0.0.  
**Se aconseja subir a e-aulas versiones parciales de la solución a la actividad.**
8. Todas las evaluaciones serán realizadas en el sistema operativo GNU/Linux.
9. Todas las entregas están sujetas a herramientas automatizadas de detección de plagio en códigos.

No habrán excepciones a estas reglas.

Solve the following problem based on what was discussed during the lectures, and your own work, on polygon triangulation and the art gallery problem. You may write pseudocode using the notation followed in class, or an enumerated brief description of the steps followed by the algorithm.

To solve the following problems always assume that the following data structures are being used.

- Status structure  $\mathcal{T}$ : balanced binary search tree, which keeps track of the line segments that intersect the sweep line  $\ell$ .
- Event queue  $\mathcal{Q}$ : priority queue, that keeps record of the relevant events handle by  $\ell$ . The number of events is static; it does not change during the sweep.
- Planar subdivision  $\mathcal{D}$ : doubly-connected edge list. This is the data structure of choice to describe planar subdivisions.

**Problem statement:**

1. [40 pts.] Consider the polygon triangulation algorithm discussed in class. The subroutine, `MAKE_MONOTONE(P)`, that partitions a simple polygon  $P$  into monotone subpolygons handles the five different types of vertices in a different way. Write down the pseudocode for each subroutine that handles each vertex and explain briefly their computational complexity in terms of the number of vertices  $n$  and the complexity of the subdivisions.
2. [30 pts.] In computational geometry, degeneracies refer to those cases where an algorithm might give wrong answers or not work at all. Discuss the degeneracies of the polygon triangulation algorithm discussed in class, and propose a way to solve them. For instance, can the algorithm be used to triangulate non-simple polygons or general planar subdivisions? If so, explain briefly why. If not, how would you change the algorithm to make it work?
3. [30 pts.] What is the most crucial part/ingredient of the polygon triangulation algorithm? To solve this question, think in terms of the Lemmas and Theorems discussed in the Dutch book and during lectures.

**Important:** Do not exceed two letter-sized pages to write down your solution.