



MIDTERM ASSIGNMENT 1

February 22, 2022

Indicaciones generales

1. Fecha de publicación: 23 de febrero de 2022 desde las 6:00.
2. Fecha de entrega: 1 de marzo de 2022 hasta las 23:59.
3. Único medio de entrega: <https://e-aulas.urosario.edu.co>.
4. Formato de entrega: módulos y archivos de prueba (`main`) en Python 3.
5. Importante: no use acentos ni deje espacios en los nombres de los archivos que cree.
6. La actividad **debe** realizarse **individualmente**.
7. Los grupos pueden consultar sus ideas con los profesores para recibir orientación; sin embargo, la solución y detalles del ejercicio debe realizarlos **individualmente**. Cualquier tipo de fraude o plagio es causa de anulación directa de la evaluación y correspondiente proceso disciplinario.
8. El grupo de trabajo debe indicar en su entrega de la solución a la actividad cualquier asistencia que haya recibido.
9. El grupo no debe consultar ninguna solución a la actividad que no sea la suya.
10. El grupo no debe intentar ocultar ningún código que no sea propio en la solución a la actividad (a excepción del que se encuentra en las plantillas).
11. Las entregas están sujetas a herramientas automatizadas de detección de plagio en códigos.
12. **e-aulas** se cerrará a la hora 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.

Introduction

The ministry of education wants to build a **server farm** that is going to keep sensitive information. The server farm must fulfill a special requirement: there can be no traces of humidity or water infiltration; otherwise data may be compromised. Therefore, the government hires a **hydrology** company to explore the construction site so they can locate all water and humidity sources; like those coming from rain.

In order to map out the terrain the company has randomly sampled different parts of it, registering *location and elevation*. The final goal is to provide information on where to locate drain pipes and design a **drainage network** so the farm can stay dry with no humidity issues that could damage the servers.

A drainage network keeps information on the hydrologic properties of the terrain; for example, water source locations and potential flood areas. The collection of water streams, sources, basins, and sinks forms such a network, see Fig. 1.

There is a problem, however. The hydrology company has no software to post-process the information and construct the whole drainage network. So they have decided to outsource this part of the job to you! It is going to be your duty, then, to solve the problem of the drainage network.

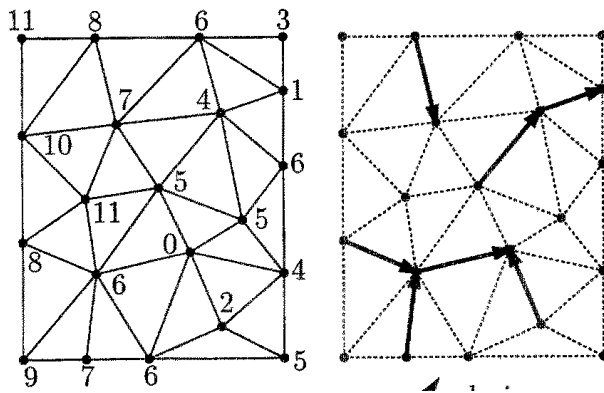


Figure 1: *Left.* Top view of the construction site, with the randomly sampled points —vertices—, modeled as a triangulated irregular network with elevations at its vertices (see below for details). *Right.* Top view of the drainage network of the construction site showing water sources, water pits, and the flow of water.

Problem

Your task is to design and implement an application, along with a test suite, that specifically addresses the functionality needed by the hydrology company. The functionality required evaluates aspects of the hydrologic features of the construction site. Such features are local properties of drainage network such as points, lines, and regions of special interest to the flow of water due to gravity and obstacles present on the terrain (see Fig. 1).

The functionality of **the application must be implemented as a class `TIN` and its corresponding test suite**, with the following specifications:

1. Plot the elevation profile using different colors for different ranges of elevations. Given the set of sample points on the terrain, plot its elevation profile as a three dimensional graph.
2. Given the *planar* location of a point find its elevation value. Do this by means of a linear interpolation using the sample data. Highlight the data points used in the interpolation.
3. Query whether a point is a water source. A water source is a peak point in the sample data of relative maximum. Report all water sources on the terrain.
4. Query whether point is a water pit. A water pit is an anti-peak point in the sampled data of relative minimum. Report all water pits on the terrain.
5. Construct and plot the dual graph of the triangulated terrain and check that it is indeed a tree. You may use any data structure for graphs defined in `scipy`.

Notice that the entry points mentioned in 2 need not be one of the sampled points in the terrain. If these points are part of the sampling it should be apparent how to proceed.

The test suite need not be implemented using unit testing, simple tests showing the correctness of the implementation should suffice. Additionally, all answers should generate a different plot thereby graphically showing the solution. Items 1, 2, 3, and 4 must generate a three-dimensional plot, 5 a two-dimensional one.

Strategy

Let us now analyze how to proceed. The input to the problem is the randomly sampled data of the terrain recorded as *location*: a point in the plane, and *elevation*: a real number.

Next, remembering what you learned in Computational Geometry, you know that you need to build a computational model of the sampled terrain so that you can efficiently process the

information. A good candidate for this is a **digital elevation model** (DEM), which is a computational model for representing terrain relief based on a finite number of points or samples. There are different types of DEMs, but since the sample points on the terrain are not uniformly distributed you know that the best DEM for the job is a triangulated irregular network, see Fig. 2.

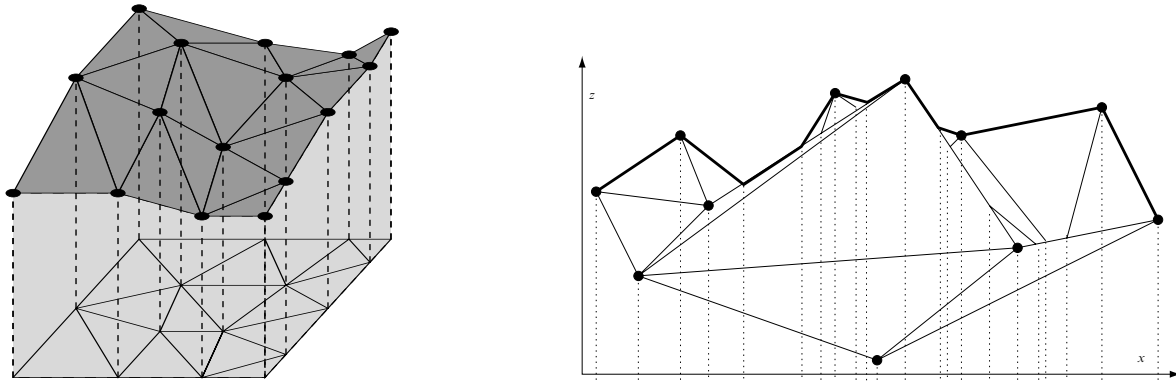


Figure 2: *Left*: Three dimensional view of the DEM of the construction site modeled as a TIN. *Right*: Side view of the DEM showing an elevation profile, ridges, and valleys.

A **triangulated irregular network** (TIN) is a **polyhedral terrain**; that is, the image of a piecewise-defined linear function with domain in two dimensions. A TIN is a triangulation of the location points, as vertices of the triangulation, augmented with the elevation data. This data structure gives rise to a piecewise-linear function that allows for interpolation of the elevations at any planar location (point), even though such location is not in the sample set, see Fig. 2.

Hence, a TIN is generated from the sampled points on the terrain by computing the Delaunay triangulation having vertices at the sample points. Thus, a TIN is constructed using algorithms for Delaunay triangulations. Having this we can work on the implementation of the application.

Now that we have a clear picture of how to proceed with the construction of the data structure, it is possible to start implementing the functionality required by the hydrology company specified above. Start by analyzing what the prototype of the class methods should look like, implement them, and finally, test them.

References

- L. De Floriani, P. Magillo, and E. Puppo. Applications of Computational Geometry to Geographic Information Systems. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, chapter 7, pages 333-388. North Holland, 2000.
- L. Floriani and P. Magillo. **Algorithms for Visibility Computation on Terrains: A Survey**. *Environment and Planning B: Planning and Design* **30**, issue 5, pages 709-728 (2003).