

# Prácticas de Algorítmica.

## 3º de Grado en Ingeniería Informática.

### Curso 2019-2020.

#### Práctica 5.

##### Objetivos.

Con esta práctica se pretende que el alumno implemente un algoritmo basado en la técnica de la programación dinámica.

##### Enunciado:

El alumno deberá implementar el algoritmo basado en programación dinámica, explicado en clase de teoría, para obtener una aproximación poligonal de una curva digital con mínimo error. En el tema de programación dinámica se han estudiado dos métodos óptimos para obtenerlas y se han visto todos los conceptos relacionados con las aproximaciones poligonales de una curva digital.

Se suministra en el material un ejemplo completo de cómo desarrollar un algoritmo para obtener aproximaciones poligonales que se explicará con detalle en clase de prácticas. Este ejemplo contiene lo siguiente:

- Clase **Point** que representa un punto en 2D.
- Clase **Line** que representa una recta en 2D.
- Clase **Matrix** que representa una matriz, necesaria para hacer cálculos auxiliares.
- Clase **DigitalCurve** que representa una curva digital compuesta de una serie de puntos. Puede ser abierta o cerrada.
- Clase **Algorithm** que es una clase abstracta de la cual ha de heredar cualquier método para obtener aproximaciones poligonales que queramos implementar. Esta clase tiene implementadas la mayoría de las funciones comunes que usará cualquier método de obtención de aproximaciones poligonales.
- Clase **SuppressionCollinearPointsMethod** que es una clase heredada de la clase Algorithm. En esta clase se implementa un método para obtener la aproximación poligonal con el menor número de puntos que produce un error 0. Este método obtiene dichos puntos usando un método de la clase Algorithm que elimina los puntos alineados (collinearPointsElimination).
- Programa principal (**main.cpp**) en el cual se muestra como usar el método implementado.
- **CmakeList.txt** para generar el makefile y el ejecutable.
- Ejemplos de curvas digitales (tienen extensión txt) para probar el programa.

El alumno deberá implementar una clase para implementar un método voraz. Dicha clase se llamará **DynamicProgrammingMethod** que el alumno tendrá que implementar en los archivos **dynamicprogrammingmethod.hpp** y **dynamicprogrammingmethod.cpp**. La forma de implementarlos es similar a la de la práctica 4.

La forma de proceder es la siguiente:

- En primer lugar, en el programa principal habrá que introducir el fichero que contiene la curva digital y cuantos puntos queremos que tenga la aproximación poligonal. Si la curva es cerrada, como ocurre en todas las curvas suministradas, el punto de comienzo y el final son iguales, por tanto esto hay que tenerlo en cuenta. Si quisiéramos obtener una aproximación de 4 puntos, habrá que introducir 5.
- Aplicar el método para obtener la aproximación poligonal óptima. El programa ha de mostrar el error óptimo de la aproximación y guardar en un fichero la aproximación poligonal óptima, para poder visualizarla con el programa gnuplot.

**Fecha de comienzo: 18 de noviembre.**

**Fecha máxima de entrega: 2 de diciembre.**