

Geometría Computacional  
Proyecto parte 01

Profesor: Adriana Ramírez Viguera

Ayudante: José Emiliano Cabrera Blancas

12 de mayo de 2014

## 1. Objetivos:

- Que el alumno implemente el algoritmo incremental para construir el algoritmo incremenetal.
- Que el alumno dibuje en *Ruby-Processing* los pasos que realiza el algoritmo incremental para construir un conjunto aleatorio de puntos dados.

## 2. Descripción

Aunque el algoritmo incremental para construir el diagrama de *voronoi* es sencillo en su descripción, la dificultad de este radica en mantener bien la estructura que se utilice para representarla.

Como ya les habia mencionado, reutilizaremos la *DCEL* para representar el diagrama de *voronoi*. La estructura que representa el diagrama de *voronoi* no solo contiene un apuntador hacia la *DCEL* que lo representa, si no también otros datos que nos ayudan a procesar el algoritmo.

Por otra parte, también se les proporcionan las funciones básicas para calcular la bisectriz perpendicular de dos puntos y posibles intersecciones con aristas.

La descripción de cada función y estructura que se agregaron para que ustedes puedan implementar el algoritmo vienen en los siguientes archivos del código fuente:

Archivos fuente

```
src/algorithms/algorithms.h
src/algorithms/algorithms.c
```

```

src/dcel/dcel.h
src/dcel/dcel.c

src/equations/equations.h
src/equations/equations.c

src/face/face.h
src/face/face.c

src/gui/voronoi.rb

src/main.c

src/points/2d_points.h
src/points/2d_points.c

src/shared_libraries/algorithms.rb
src/shared_libraries/voronoi.rb

src/tests/voronoi_tests.c
src/tests/voronoi_tests.h
src/tests/voronoi_tests.rb

src/voronoi/voronoi.h
src/voronoi/voronoi.c

```

### 3. *Tests*

Su código debe de pasar los siguientes *tests*:

#### *Tests*

algorithms.h: Voronoi de solo 2 puntos	[FAIL]
algorithms.h: Voronoi de 3 puntos del caso degenerado	[FAIL]
algorithms.h: Voronoi de 7 puntos aleatorios	[FAIL]

El primer *test* verifica que la función *voronoi\_incremental()* genere un diagrama de *voronoi* correcto para dos puntos.

El segundo *test* verifica que la función *voronoi\_incremental()* genere un diagrama de *voronoi* correcto para el caso degenerado de solo 3 puntos.

El último *test* verifica que la respuesta que genera la función *voronoi\_incremental()* sea correcta para cualquier conjunto de siete puntos generados de forma aleatoria.

#### 4. ¿Qué debes de programar y dónde comentar tus cambios?

Basicamente debes de completar las funciones que se te indican en el archivo *src/algorithms/algorithms.h* y el archivo *gui/voronoi.rb* que sirve para dibujar la *DCEL*.

Para dibujar la *DCEL* debes de leer el archivo que genera la función *process\_voronoi()*, dicho archivo es *src/salida.txt*.

El formato que tiene el archivo es el siguiente:

*salida.txt*

```
Semillas: 0
Aristas: 8
400.000000 0.000000 0.000000 0.000000
0.000000 0.000000 400.000000 0.000000
0.000000 0.000000 0.000000 400.000000
0.000000 400.000000 0.000000 0.000000
0.000000 400.000000 400.000000 400.000000
400.000000 400.000000 0.000000 400.000000
400.000000 400.000000 400.000000 0.000000
400.000000 0.000000 400.000000 400.000000
Semillas: 1
65.000000 5.000000
Aristas: 8
400.000000 0.000000 0.000000 0.000000
0.000000 0.000000 400.000000 0.000000
0.000000 0.000000 0.000000 400.000000
0.000000 400.000000 0.000000 0.000000
0.000000 400.000000 400.000000 400.000000
400.000000 400.000000 0.000000 400.000000
400.000000 400.000000 400.000000 0.000000
400.000000 0.000000 400.000000 400.000000
```

Donde *Semillas: N* nos indica cuantos vertices semilla se deben de dibujar y *Aristas: M* indica que aristas debemos dibujar hasta ese momento.

Por último, como ya les habia dicho en clase, el archivo *README.txt* debe contener tres pseudocódigos, ya sea descritos solo con palabras o alguna notación fácil de leer. El primer pseudocodigo debe describir los pasos que hicieron para completar la función *merge\_faces()*, el segundo pseudocodigo debe describir los pasos que realiza *voronoi\_incremental()* y el tercer pseudocodigo debe contener los pasos de la función *upgrade\_voronoi\_diagram()* para actualizar agregar o actualizar la *DCEL*.

Estos pseudocodigos son requisitos indispensables para calificar su proyecto final.

**Entrega 6 de junio de 2014**