

# Tópicos em Geometria Computacional

## Segundo Trabalho

22 de junho de 2021  
(atualizado em 1 de julho de 2021)

### 1 Introdução

O trabalho consiste em fazer uma implementação para calcular o *Diagrama de Voronoi* de um conjunto de pontos do plano.

### 2 Resolução do problema

A resolução do problema, ou seja, a descrição do problema, do algoritmo e sua corretude, deve estar em um texto claro em formato de um artigo e em pdf (com nome `relatorio.pdf`). Deve conter o nome do autor (aluno), uma introdução com o problema, o algoritmo e sua explicação. Todas as referências que forem usadas devem estar citadas corretamente no texto.

### 3 Especificação da implementação

A implementação pode ser feita em qualquer linguagem, contanto que seja possível rodar no ambiente computacional do DINF.

A entrada de dados deve ser feita pela entrada padrão (stdin) e a saída de dados pela saída padrão (stdout), ou seja, o seu programa lê do teclado e escreve na tela. O objetivo é que seja executado com redirecionamento de arquivos, como o comando abaixo:

```
$ voronoi < entrada.txt > saida.txt
```

O trabalho deve ser entregue com um `makefile` de forma que ao digitar o comando `make` o executável seja construído.

## 4 Entrega

A entrega deve ser feita por e-mail para `andre@inf.ufpr.br`, com um arquivo `tar.gz` (compactado) enviado como anexo de uma mensagem com assunto “geocomp-trabalho 2” (exatamente).

O arquivo compactado deve ter nome `fulano.tar.gz`, onde `fulano` deve ser substituído pelo “login name” do autor.

O arquivo `fulano.tar.gz`, uma vez expandido, deve criar (no diretório corrente) os arquivos abaixo:

- `makefile` (ou `Makefile`)
- `relatorio.pdf`
- e os arquivos fontes necessários para a geração do executável.

## 5 Diagrama de Voronoi

Neste trabalho o objetivo é construir o Diagrama de Voronoi de um conjunto de pontos dado. O nome do programa executável deve ser `voronoi`. A saída será uma subdivisão do plano na estrutura DCEL, onde cada face tem o mesmo índice do ponto da entrada contido na face.

Considere que todas as coordenadas dadas são números inteiros entre 1 e 99 (inclusive). Para evitar as faces ilimitadas, considere um retângulo envolvente (*bounding box*) fixo, ortogonal aos eixos e com vértice inferior esquerdo (0,0) e vértice superior direito (100,100).

A entrada de dados será um texto com a lista de pontos. A primeira linha tem um número,  $n$ , de pontos no plano. As  $n$  linhas seguintes tem as coordenadas  $X$  e  $Y$  de cada um dos pontos, separadas por um espaço em branco.

Exemplo (comentários não fazem parte do arquivo):

```
5      # número de pontos
20 50  # coordenadas  $x$  e  $y$  do 1o ponto
50 20  # ...
50 50
50 80
80 50
```

A saída de dados deve ser um texto com a descrição da estrutura de dados que representa a subdivisão encontrada em uma DCEL. A primeira linha 3

números,  $v$ ,  $a$  e  $f$ , separados por espaços, representando, respectivamente, o número de vértices, arestas e faces (sem contar a face externa) da subdivisão (contando com os vértices e arestas do retângulo envolvente).

As próximas  $v$  linhas tem as informações de cada vértice (um vértice por linha). Para cada vértice temos 3 números separados por espaço, as duas coordenadas e o índice de uma semi-aresta que tem este vértice como origem.

Em seguida temos  $f$  linhas com as informações das faces. Cada face tem as duas coordenadas do ponto (do conjunto de entrada) contido nesta face e o índice de uma semi-aresta que tem esta face como face esquerda.

Logo após temos  $2a$  linhas, uma para cada semi-aresta. Em cada linha temos 5 números separados por espaços. Os números representam, respectivamente, o índice do vértice origem, o índice da semi-aresta simétrica, o índice da face esquerda, o índice da próxima semi-aresta circulando a face esquerda, e o índice da semi-aresta anterior circulando a face esquerda. O índice da face externa deve ser 0 (zero).

Exemplo referente ao exemplo de entrada (comentários não fazem parte do arquivo):

```

8 12 5      # valores de  $v$ ,  $a$  e  $f$ 
0 0 1      # 1o vértice
0 100 2     # 2o vértice
100 0 6     # ...
100 100 4
35 35 10    # 5o vértice
35 65 12
65 35 16
65 65 18    # 8o vértice
20 50 2     # 1a face
50 20 8     # 2a face
50 50 12    # ...
50 80 4
80 50 6     # 5a face
1 2 0 3 7   # 1a semi-aresta
2 1 1 9 13  # 2a semi-aresta
2 4 0 5 1   # ...
4 3 4 14 21
4 6 0 7 3   # 5a
3 5 5 22 24
3 8 0 1 5
1 7 2 23 10

```

1 10 1 11 2  
 5 9 2 8 16    # 10a  
 5 12 1 13 9  
 6 11 3 15 19  
 6 14 1 2 11  
 2 13 4 20 4  
 5 16 3 17 12    # 15a  
 7 15 2 10 23  
 7 18 3 19 15  
 8 17 5 24 22  
 8 20 3 12 17  
 6 19 4 21 14    # 20a  
 8 22 4 4 20  
 4 21 5 18 6  
 3 24 2 16 8  
 7 23 5 6 18    # 24a semi-aresta