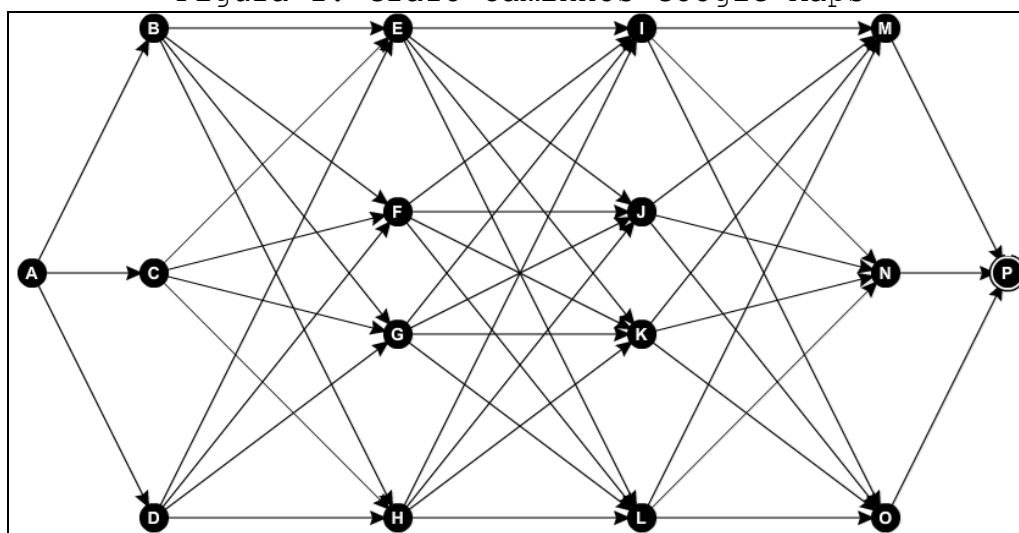




O Google Maps, segundo o próprio site, é um dos melhores aplicativos para mapeamento de rotas. Qual a relação com a disciplina? Bem, você deverá desenvolver um programa que visa mostrar os caminhos mais curtos entre a sua localização e a localização destino do Google Maps. Suponha o grafo da Figura 1, onde estamos localizados no vértice A e o destino no vértice P; o objetivo é encontrar o(s) caminho(s) mais curto(s) para que o usuário possa optar pelo caminho mais curto.

Figura 1: Grafo Caminhos Google Maps




Fonte: elaborado pelo autor.

O grafo pode ser representado através de uma matriz de adjacência, conforme apresentado na Figura 2.

Figura 2: Matriz de Adjacência

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
A	0	2	1	3	0	0	0	0	0	0	0	0	0	0	0	0
B	2	0	0	0	7	2	5	3	0	0	0	0	0	0	0	0
C	1	0	0	0	5	4	2	8	0	0	0	0	0	0	0	0
D	3	0	0	0	3	8	3	5	0	0	0	0	0	0	0	0
E	0	7	5	3	0	0	0	0	10	15	8	5	0	0	0	0
F	0	2	4	8	0	0	0	0	6	8	4	9	0	0	0	0
G	0	5	2	3	0	0	0	0	8	2	17	13	0	0	0	0
H	0	3	8	5	0	0	0	0	3	5	5	11	0	0	0	0
I	0	0	0	0	10	6	8	3	0	0	0	0	2	3	6	0
J	0	0	0	0	15	8	2	5	0	0	0	0	1	6	8	0
K	0	0	0	0	8	4	17	5	0	0	0	0	5	5	5	0
L	0	0	0	0	5	9	13	11	0	0	0	0	8	4	4	0
M	0	0	0	0	0	0	0	0	2	1	5	8	0	0	0	3
N	0	0	0	0	0	0	0	0	3	6	5	4	0	0	0	5
O	0	0	0	0	0	0	0	0	6	8	5	4	0	0	0	4
P	0	0	0	0	0	0	0	0	0	0	0	0	3	5	4	0

Fonte: elaborado pelo autor.

	Análise e Desenvolvimento de Sistemas Ciência da Computação Sistemas de Informação	2017/2 Trabalho GA (3,0) Laboratório II
	Prof. Gilberto Irajá Müller	

Ao observar a Figura 2, podem-se fazer as seguintes conclusões:

- Do vértice A é possível ir apenas para os vértices B, C e D;
- Em relação à distância: A até B é 2; A até C é 1 e A até D é 3; considera-se a distância uma unidade qualquer (metros, quilômetros, etc.);
- Ressalta-se ainda que uma matriz de adjacência é espelhada, ou seja, os valores podem ser obtidos tanto a partir da linha como a partir da coluna; como se fôssemos dobrar uma folha;
- As linhas e colunas da matriz $(0, 1, \dots, n - 1, n)$ correspondem as letras dos vértices: (A, B, ..., O, P).

Importante: A relação entre os pontos deve seguir uma regra de precedência, ou seja, todos os pontos daquele nível deverão estar conectados com todos os pontos do nível seguinte. Ex.: os pontos B, C e D estão conectados com todos os pontos seguintes (E, F, G e H).

Estruturas de Dados


PASSO 1) Você deverá criar a seguinte interface, pois é o contrato obrigatório para a execução da classe de teste.

```
public interface IGoogleMaps {
    public void loadMatrix(File file) throws IOException;
    public Chromosome generateChromosome();
    public void insertOrderedChromosome(Chromosome chromosome);
    public int distance(int[] genes);
    public void evolve();
    public void print();
}
```

PASSO 2) Deverá ser criada uma classe denominada GoogleMaps que implementa a interface IGoogleMaps conforme Quadro 1.

Quadro 1: Descrição dos Contratos

Método/operação	Descrição
loadMatrix	Responsável pelo carregamento da matriz de adjacência constante em arquivo csv (ver ANEXO A).
generateChromosome	Responsável por gerar um novo cromossomo de forma aleatória (ver ANEXO B).
insertOrderedChromosome	Responsável pela inserção ordenada em um array de "melhores" cromossomos.

 UNISINOS UNIVERSIDADE DO VALE DO RIO DOS SINOS	Análise e Desenvolvimento de Sistemas Ciência da Computação Sistemas de Informação	2017/2 Trabalho GA (3,0) Laboratório II
	Prof. Gilberto Irajá Müller	

	Guardaremos os 10 melhores cromossomos em ordem de distância (crescente) neste array. Se o cromossomo a ser inserido é ruim aos que já estão no array e o array está todo preenchido, então, este deverá ser desconsiderado. Se o cromossomo já consta no array, então, deverá ser desconsiderado. Caso seja um cromossomo bom e o array esteja completo, então, o pior cromossomo deverá ser retirado da lista, sendo que o novo cromossomo deverá ser inserido. Caso o array não esteja cheio, então, deverá ser inserido conforme sua distância (ou seja, qualidade). Poderão existir cromossomos com o mesmo fitness (critério de aptidão/qualidade).
distance	Responsável por calcular a distância total do cromossomo, ou seja, o fitness. Observa-se que o parâmetro refere-se aos genes do cromossomo. Utiliza-se a matriz de adjacência no cálculo da distância.
evolve	Responsável pela evolução/seleção dos melhores cromossomos. Faremos 100 iterações, onde cada iteração irá gerar um cromossomo e este inserido no array de melhores cromossomos.
print	Responsável pela impressão dos 10 melhores cromossomos, do menor caminho para o maior caminho.

Fonte: elaborado pelo autor.

PASSO 3) Abaixo, segue estrutura da classe de teste:

```

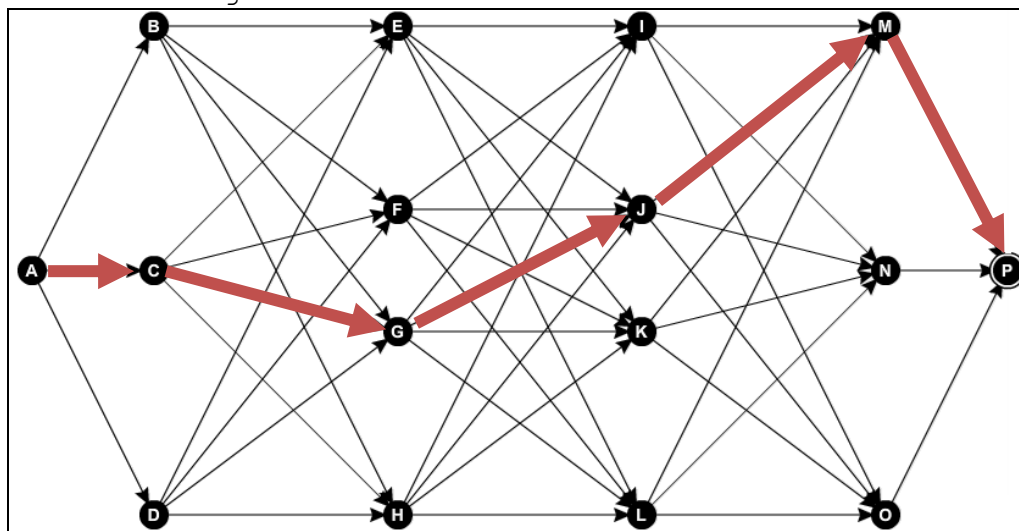
public class GoogleMapsTest {
    public static void main(String[] args) {
        IGoogleMaps googleMaps = new GoogleMaps();
        try {
            googleMaps.loadMatrix(new File("ARQUIVO MATRIZ ADJ.));
        } catch (IOException e) {
            e.printStackTrace();
        }
        googleMaps.evolve();
        googleMaps.print();
    }
}

```

O resultado da execução do GoogleMapsTest deve ser algo similar ao que está abaixo. A Figura 3 ilustra o primeiro caminho.

```
Short path -> 9 A C G J M P
Short path -> 12 A D G J M P
Short path -> 13 A B G J M P
Short path -> 13 A B H I M P
Short path -> 14 A B H J M P
Short path -> 16 A C G J N P
Short path -> 16 A C G I M P
Short path -> 16 A C F I M P
Short path -> 17 A C H I M P
Short path -> 18 A B F K N P
```

Figura 3: Caminho Curto Distância 9




Fonte: elaborado pelo autor.

ANEXOS

ANEXO A: Você deverá criar na classe GoogleMaps uma matriz (linha vs. coluna) de inteiros que armazenará a matriz de adjacência obtida a partir da leitura de um arquivo e um atributo inteiro para armazenar o tamanho do gene. Abaixo, exemplo de conteúdo do arquivo separado por ponto-vírgula, onde a primeira linha representa o tamanho do gene e, as demais, as distâncias entre os pontos. O tamanho do array deverá ser definido a partir dessa leitura.

```
6
0;2;1;3;0;0;0;0;0;0;0;0;0;0;0;0
2;0;0;0;7;2;5;3;0;0;0;0;0;0;0;0
1;0;0;0;5;4;2;8;0;0;0;0;0;0;0;0
```

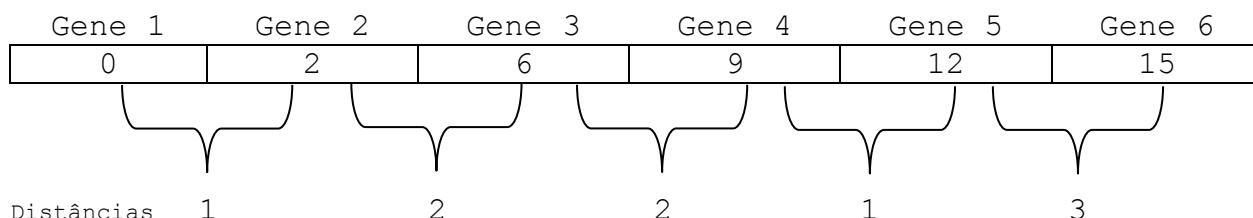
	Análise e Desenvolvimento de Sistemas Ciência da Computação Sistemas de Informação	2017/2 Trabalho GA (3,0) Laboratório II
	Prof. Gilberto Irajá Müller	

```

3;0;0;0;3;8;3;5;0;0;0;0;0;0;0;0;0
0;7;5;3;0;0;0;0;10;15;8;5;0;0;0;0;0
0;2;4;8;0;0;0;0;6;8;4;9;0;0;0;0;0
0;5;2;3;0;0;0;0;8;2;17;13;0;0;0;0;0
0;3;8;5;0;0;0;0;3;5;5;11;0;0;0;0;0
0;0;0;0;10;6;8;3;0;0;0;0;2;3;6;0;0
0;0;0;0;15;8;2;5;0;0;0;0;1;6;8;0;0
0;0;0;0;8;4;17;5;0;0;0;0;5;5;5;0;0
0;0;0;0;5;9;13;11;0;0;0;0;8;4;4;0;0
0;0;0;0;0;0;0;0;2;1;5;8;0;0;0;0;3
0;0;0;0;0;0;0;0;3;6;5;4;0;0;0;0;5
0;0;0;0;0;0;0;0;6;8;5;4;0;0;0;0;4
0;0;0;0;0;0;0;0;0;0;0;0;0;3;5;4;0

```

ANEXO B: Para gerar um novo caminho, iremos utilizar um conceito básico da genética que é o de cromossomo¹ e gene. Um cromossomo é composto por vários genes e, no exemplo acima, teremos "6" genes (primeira linha do arquivo) em função do comprimento entre o vértice A (nossa localização) e o vértice P (localização destino). Deve-se utilizar um array unidimensional de inteiros que são os genes e representam o cromossomo. Exemplo de um cromossomo:




A distância total é **9** e o caminho: **A -> C -> G -> J -> M -> P**.

Como gerar um novo cromossomo? Imagine que a coluna à esquerda da tabela abaixo seja o índice do array do nosso cromossomo, então, para os índices 1 a 4, iremos gerar números aleatórios que representam as opções possíveis daquele vértice no grafo. Observe que poderá variar o número de possibilidades e o número de genes.

Index

0	Sempre será 0 (vértice A - origem).
1	Sortear valores entre 1 (B), 2 (C) ou 3 (D).
2	Sortear valores entre 4 (E), 5 (F), 6 (G) ou 7 (H).
3	Sortear valores entre 8 (I), 9 (J), 10 (K) ou 11 (L).
4	Sortear valores entre 12 (M), 13 (N) ou 14 (O).
5	Será 15 (vértice P - último vértice que é o destino).

¹ Leia-se cromossomo bom para aquele que tem um caminho curto. Leia-se cromossomo ruim para aquele com um caminho longo.

	Análise e Desenvolvimento de Sistemas Ciência da Computação Sistemas de Informação	2017/2 Trabalho GA (3,0) Laboratório II
	Prof. Gilberto Irajá Müller	

Após a geração do cromossomo, deverá ser calculada a distância armazenando-a no atributo fitness do cromossomo. A Figura 4 apresenta a classe de sugestão para o cromossomo.

Figura 4: Estrutura Cromossomo

Chromosome
- genes : int[] - fitness : int
+ Chromosome() + Chromosome(genes : int[], fitness : int) + getGenes() : int[] + setGenes(genes : int[]) : void + getFitness() : int + setFitness(fitness : int) : void + toString() : void

powered by Astah

Fonte: elaborado pelo autor.

Avaliação

- Comentários em todas as classes/métodos;
- O código deverá ser claro (coloque nomes que mostre a intenção da ação);
- Não será aceito a apresentação do trabalho após a data limite; observa-se que a nota será considerada apenas mediante apresentação;
- Não será aceito trabalho igual ao de outros colegas ou cuja solução tenha sido copiada da internet ou de livros. Esta prática é chamada de plágio;
- Não deverá ser utilizada outra estrutura de dados que não seja as apresentadas no trabalho;
- O trabalho deve ser apresentado para o professor em aula.