

---

Aluno:	Matrícula:	Valor: 25,0	Nota:
--------	------------	-------------	-------

---

## Trabalho I

### 1. Objetivos.

- Desenvolver a habilidade de programação de algoritmos em grafos.
- Reforçar o aprendizado sobre algoritmos para o problema do caixeiro viajante.
- Exercitar as competências em metodologia e escrita científica.
- Identificar vantagens e desvantagens de diferentes estratégias computacionais para problemas difíceis.

### 2. Descrição.

O trabalho consiste em desenvolver algoritmos para solucionar o Problema do Caixeiro Viajante (PCV). Para tal, deve-se utilizar a biblioteca desenvolvida no Trabalho I. Os algoritmos a serem desenvolvidos dividem-se em duas categorias:

**Algoritmos Construtivos** Cujas finalidades é criar uma rota inicial para o problema do caixeiro viajante.

**Algoritmo de Refinamento** cujo objetivo é melhorar a rota gerada pelo algoritmo anterior examinando alternativas de troca de caminhos.

Há uma vasta gama de algoritmos, tanto construtivos, quanto de refinamento para o PCV. Ademais, detalhes de implementação podem fazer toda a diferença na eficiência dos algoritmos. Assim, os trabalhos submetidos participarão de uma competição, onde os mais eficientes receberão as melhores notas no critério desempenho. A competição ocorrerá da seguinte maneira: cada trabalho será executado por 1 minuto em cada um dos grafos de entrada. O trabalho que conseguir o menor custo nesse grafo ficará em 1º lugar, o de segundo menor custo em 2º e assim por diante. Aquele que obtiver a média mais baixa das colocações será o campeão da competição de algoritmos para o PCV e receberá a nota máxima nesse critério. Os demais receberão notas proporcionais ao seu desempenho até o mínimo de 60% da nota.

Dentre os algoritmos construtivos podemos ressaltar:

- Criação aleatória das rotas.** Simplesmente cria uma permutação aleatória dos vértices a serem visitados e adiciona o vértice origem ao final para completar o ciclo.
- Vizinho mais próximo.** A cada iteração caminha para o vértice mais próximo que ainda não foi visitado até que todos os vértices façam parte da rota.
- Inserção mais barata.** Selecione uma aresta arbitrária. A cada iteração, calcule o custo de inserção de cada vértice a essa rota. Adicione o vértice que gera a rota de menor custo. Encerre quando a rota conter todos os vértices.

Já para o refinamento, os métodos mais comuns são:

- 2-opt.** Remove duas arestas distintas da rota e as reinsere da melhor forma possível.

- (b) **3-opt**. Remove três arestas distintas da rota e as reinsere da melhor forma possível.
- (c) **k-opt**. Remove  $k$  arestas distintas da rota e as reinsere da melhor forma possível. Idealmente executa-se esse refinamento para valores crescentes de  $k = \{2, 3, 4, \dots\}$ .

Deve-se implementar ao menos um algoritmo construtivo e um algoritmo de refinamento para solucionar o problema. Pode-se inclusive implementar outros algoritmos que não foram aqui citados na busca por melhores algoritmos.

Seu programa deve perguntar o nome do grafo de entrada e o tempo (em segundos) para execução dos algoritmos:

```
1 > Informe o grafo: teste.txt
2 > Tempo limite (s): 60
```

processar o algoritmo e escrever a saída em um arquivo da seguinte forma:

```
1 16.3
2 0 1 5 4 3 2 0
```

onde a primeira linha representa o custo da rota e a segunda linha, a sequência de vértices da rota.

Adicionalmente, deve-se **redigir um artigo** contendo: (i) uma introdução sobre o problema do caixeiro viajante; (ii) o pseudocódigo e explicações sobre os algoritmos implementados; (iii) os resultados dos custos das rotas obtidas pelos algoritmos nos grafos de teste; e (iv) as principais conclusões sobre os algoritmos desenvolvidos. O artigo deve estar formatado conforme o modelo da Sociedade Brasileira de Computação<sup>1</sup>.

### 3. Condução de Experimentos.

Cada trabalho deverá executar o algoritmo desenvolvido sobre o conjunto de grafos disponíveis no arquivo “Datasets.rar” com um tempo limite de 60 segundos. Como há componentes de aleatoriedade nos algoritmos, sugiro executar ao menos 5 vezes em cada grafo e reportar o custo médio e o custo da melhor rota obtida no artigo. Esses grafos são provenientes de aplicações reais dos problemas considerados, tais como alocação de horários, alocação de registradores, jogos etc. A tabela abaixo apresenta uma breve descrição sobre cada arquivo de teste:

Arquivo	Descrição
teste.txt	Grafo artificial cuja rota ótima é trivial.
a280.txt	Problema de perfuração.
ali535.txt	535 aeroportos ao redor do mundo.
ch130.txt	Rota entre 130 cidades.
fl1577.txt	Manufatura de circuito VLSI com 577 pontos.
gr666.txt	Rota entre 666 cidades ao redor do mundo.

Sugiro utilizar a instância “teste.txt” para testar a corretude dos algoritmos visto que é fácil verificar seu resultado manualmente.

### 4. Avaliação.

O trabalho deverá ser feito **individualmente ou em dupla** e enviado via Moodle (código e artigo) até as 23:59h do dia 25/08/21. A distribuição dos pontos será a seguinte:

<sup>1</sup><https://www.sbc.org.br/documentos-da-sbc/send/169-templates-para-artigos-e-capitulos-de-livros/878-modelosparapublicaodeartigos>

- Desempenho e corretude dos algoritmos - 15 pontos.
- Qualidade do artigo - 10 pontos.

De modo a começar um portfólio dos trabalhos que vocês desenvolveram na graduação, sugiro criar uma conta no <https://github.com/> e postar os códigos do trabalho nessa plataforma. Esse recurso tem sido cada vez mais usado para recrutar pessoas em tecnologia.

Bom trabalho!