

---

Aluno:	Matrícula:	Valor: 20,0	Nota:
--------	------------	-------------	-------

---

## Trabalho II

### 1. Objetivos.

- Verificar aplicações de teoria de grafos em problemas reais.
- Promover a familiarização dos estudantes com a pesquisa em grafos/algoritmos.
- Desenvolver conhecimentos sobre algoritmos heurísticos e problemas combinatoriais em grafos.

### 2. Descrição.

O trabalho consiste em desenvolver um algoritmo heurístico para solucionar o problema do caixeiro viajante. Esse é um clássico problema pertencente à classe  $\mathcal{NP}$  e, como foi discutido na disciplina, não é possível solucionar em tempo viável problemas dessa classe através de algoritmos de enumeração, assim bons algoritmos heurísticos são mais aplicáveis nesses casos. A implementação do algoritmo heurístico será feita em duas etapas:

- (a) Algoritmo construtivo: cria uma solução inicial de acordo com um pseudocódigo heurístico;
- (b) Refinamento: experimenta trocas de rotas de modo a melhorar a qualidade da solução gerada pelo algoritmo construtivo.

O retorno do algoritmo para o Caixeiro Viajante será uma lista informando a ordem em que as cidades (vértices) serão visitadas. Os pseudocódigos das duas funções a serem implementadas são dados nos slides 33 e 34 da aula A09. Para o *tempoDisponivel* no slide 34 considere 60 segundos. Seu programa deve ler o nome do arquivo do grafo, informar a distância total obtida pelo algoritmo construtivo Nearest Neighbor, aplicar a heurística de refinamento 2-opt à solução obtida pelo Nearest Neighbor e informar a distância final, a rota obtida e o tempo de execução. Ex.:

Ex.:

```
Grafo: <toy.txt>
Distância NN: 17
Processando ...
Distância 2opt: 16
Rota: [0, 3, 4, 2, 1, 0]
Tempo: 60.003ms
```

### 3. Condução de Experimentos.

Cada trabalho deverá validar sua implementação sobre os conjuntos de dados disponíveis no arquivo “Datasets.rar”. Esses conjuntos de dados 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
toy.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.
gr666.txt	Rota entre 666 cidades ao redor do mundo.
xql662.txt	Manufatura de circuito VLSI com 662 pontos.

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

### 4. Avaliação.

O trabalho deverá ser feito individualmente. O relatório e o código fonte deverão ser enviados via Moodle até as 23:55 horas do dia 20/11/19 para turma a T22 e 21/11/19 para a turma T21. Nesse mesmo dia o professor realizará ainda uma arguição individual aos alunos com relação à implementação do algoritmo. Pontos a serem considerados:

- Corretude e arguição sobre as implementações - 15 pontos.
- Presença / comprometimento nas aulas de laboratório - 5 pontos.

### 5. Relatório.

**Caixeiro Viajante:** Marque abaixo o distância total percorrida na solução encontrada pelo algoritmo NearestNeighbor e depois do refinamento pela função TwoOpt.

Arquivo	NearestNeighbor	TwoOpt	Melhor solução
toy.txt	17	16	16
a280.txt			?
ali535.txt	2671.06833699999	2523.18896899999	?
ch130.txt	7575.28629299999	7398.35353299999	?
gr666.txt	4110.899869	4073.85327600000	?
xql662.txt	3244.80696800000	3205.15886600000	2513

Bom trabalho!