

➤ Objetivo do Trabalho:

Implementar (**Valor 1,0**) na Linguagem C ou C++ a Metaheurística estudada no trabalho T1 para solucionar o **Problema do Caixeiro Viajante Simétrico (PCVS)**, que é um dos mais conhecidos e estudados na área de Otimização Combinatória.

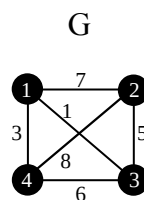
Seja um **grafo simples completo não-orientado** com n vértices e m arestas, onde o custo d_{ij} de cada aresta (i, j) representa a distância da cidade i para a cidade j ($1 \leq i, j \leq n$). Considere ainda que $d_{ij} = d_{ji}$. Ou seja, a distância da cidade i para a cidade j é a mesma da cidade j para a i .

Assim, o objetivo do PCVS é partir de um dos vértices (chamado de Origem), passar por todos os demais $n-1$ vértices sem repeti-los e retornar para o vértice Origem, fazendo isto com a menor distância (custo) possível. Repare que o vértice Origem é repetido somente duas vezes, na saída e na chegada, e que os demais vértices não permitem repetição.

IMPORTANTE

- O grupo tem que preparar um **texto** apresentando **detalhadamente** a **estratégia** que o algoritmo utilizou para resolver o problema (adicionando um exemplo da execução à escrita da explicação da estratégia) e como executar o algoritmo;
- O algoritmo tem que ter como entrada um arquivo .txt que contém na primeira linha o **número de vértices** e o **número de arestas** do grafo (separados por espaço). As demais linhas armazenam a **matriz distância D** , onde o elemento d_{ij} armazena a distância da cidade i para a cidade j ($1 \leq i, j \leq n$). Veja o exemplo a seguir do arquivo .txt que representa o grafo G abaixo:

4	6
0	7 1 3
7	0 5 8
1	5 0 6
3	8 6 0



Obs.: Para este trabalho, considere o vértice 1 como o vértice Origem.

- O **texto** (.pdf) e o **código do algoritmo** têm que ser enviados até às **23h59** do dia **20/09/2023** (quarta-feira) via e-mail para:

philippeal@yahoo.com.br

- O grupo realizará a **apresentação oral (Valor 0,5) do algoritmo** na aula do dia **22/09/2023**, quando **todos os componentes do grupo devem estar presentes** e o Professor poderá realizar perguntas sobre o código para qualquer componente;
- Caso algum componente do grupo **não esteja presente na aula de apresentação do trabalho**, o mesmo ficará **sem os pontos do código** (Valor 1,0) e **da apresentação** (Valor 0,5);
- Caso o grupo **somente envie o código e não realize a apresentação do trabalho**, **todos os componentes do grupo ficarão sem os pontos do código** (Valor 1,0) e **da apresentação** (Valor 0,5);

O **ASSUNTO** do e-mail a ser enviado deve ter a seguinte formatação:

PAA-T2-PrimeiroNome dos Alunos

Exemplo de e-mail do grupo (fictício) formado pelos alunos Lucas Pereira, Renato da Silva e Miguel dos Santos:

Para: philippeal@yahoo.com.br

De: Lucas Pereira

Assunto: PAA-T2-Lucas-Renato-Miguel

Anexo: PAA-T2-Lucas-Renato-Miguel.zip

Repare que o nome do arquivo compactado deve ter a mesma formatação do “Assunto” do e-mail:

PAA-T2-Lucas-Renato-Miguel.zip

- E-mails com o **Assunto** e **Nome do Arquivo** fora do padrão **NÃO SERÃO ACEITOS**;

- Funções têm que ser utilizadas no código;
- Não deixe para enviar o trabalho na última hora, pois podem acontecer problemas com o envio;
- Após a hora e a data marcada para o envio do trabalho, **NÃO É MAIS PERMITIDO ENVIÁ-LO**;
- O e-mail com o trabalho considerado para correção será o **ÚLTIMO** recebido pelo Professor, **dentro do prazo determinado**;
- Apenas um e-mail por grupo deve ser enviado. Isto é, não há a necessidade de cada componente do grupo enviar. **No entanto, é recomendável enviar uma cópia do e-mail para os demais componentes do grupo.**
- Preste atenção se está enviando a versão correta do trabalho, visto que será considerada a data da entrega da versão correta;
- Algoritmos que não estão compilando não serão aceitos. Portanto, não é necessário enviar, receberão nota **ZERO**. Caso o grupo esteja com dúvidas, tire-as com o Professor antecipadamente. **Não deixe para a última hora**;
- A primeira linha de cada código deve conter os nomes dos **componentes do grupo**;
- Utilize nomes sugestivos para as variáveis. Faça corretamente a indentação e comentários no código para facilitar seu entendimento;
- A chave de fechamento de um comando deve estar na mesma direção do comando que fez a abertura. Por exemplo:

```
if(x > 0){
```

```
} //O fechamento deve estar alinhado ao comando.
```

- **Os algoritmos serão avaliados/testados no Ubuntu 22.04.2 com gcc versão 11.4.0.** Caso o grupo faça o algoritmo no Windows, **recomenda-se fortemente testá-lo no Ubuntu** para que o mesmo não apresente problemas durante a sua avaliação. **Qualquer problema apresentado pelo algoritmo durante a avaliação é de inteira responsabilidade do grupo.**