

# Trabalho Final Fundamentos de Projeto e Análise de Algoritmos

## Integrantes:

Thiago Mariano  
Felipe Espindola  
Julia Evelyn  
Matheus Nolasco

O trabalho busca estudar e utilizar tipos de algoritmos estudados durante o semestre durante as aulas de Fundamento de Projeto e Análise de Algoritmo.

### Problema 01 Rotas de Caminhão:

Parâmetros utilizados como teste da modelagem e algoritmo desenvolvido:

Rotas (km) : 80 Km, 25 Km, 60 Km, 50 Km, 5 Km, 11 Km, 31 Km, 33 Km, 100 Km;  
Número de caminhões: 05 caminhões;

### Algoritmo guloso

O algoritmo guloso é baseado na escolha rápida e máxima de preenchimento da resposta com o objeto atual, tendo seu limite de acordo com a lógica do problema. Dessa forma, como o problema não possui um teto máximo da rota de cada caminhão, o algoritmo irá preencher todo o primeiro caminhão com a quilometragem de todas rotas, já que como dito anteriormente não há um gargalo que faça-o desistir dessa opção, porém como o objetivo é melhor distribuir as rotas entre os caminhões, **o algoritmo guloso sempre irá falhar na resolução desse problema.**

**Tendo seu tempo médio de execução em 1 milésimo de segundo**, já que não há consideração de looping e atribuições complexas dentro do algoritmo, tendo como complexidade (N) já que passar por todos caminhões.

Segue execução final do código:

```
Caminhao: Caminhao1 total Km: 395
Caminhao: Caminhao2 total Km: 0
Caminhao: Caminhao3 total Km: 0
Caminhao: Caminhao4 total Km: 0
Caminhao: Caminhao5 total Km: 0
Tempo de execucao: 1
```

## Algoritmo BackTracking

O algoritmo BackTracking gera uma sequência de possibilidades de inserção dentro da solução, como uma árvore e em forma de busca em profundidade, podando os ramos possíveis que ele visualizar que não irão agregar dentro da solução, o que evita diversos loopings que não vão agregar na solução e assim reduzimos a complexidade do algoritmo. No caso ele irá se basear unicamente em diminuir a diferença entre o caminhão com maior quilometragem de rota e o menor, logo ele irá verificar se colocar aquela rota no caminhão indicado no loop irá ajudar a diminuir a diferença, caso melhore a solução ele insere, caso não ele passa para o próximo caminhão, caso ele não encontre a possibilidade de melhorar ele irá inserir no caminhão de menor rota já existente.

Em comparação com o método guloso se mostra bem mais eficiente já que agora conseguimos nos basear no **gargalo de manter dentro da menor diferença possível entre o de maior e menor quilometragem**, o que permite o algoritmo a solucionar o problema.

Tendo seu tempo médio de execução em **12 milésimo de segundos**, o que já é considerável tendo em visto a pequena quantidade de elementos utilizados na questão, porém o resultado se mostra consistente e correto:

```
Tempo de execucao: 12
Caminhao1: 80
Caminhao2: 80
Caminhao3: 71
Caminhao4: 64
Caminhao5: 100
```