

## **Projeto 1 - Melhorando as rotas do FoodDelivery**

Olá deli, espero que o seu processo de boarding nessas últimas semanas tenha sido o melhor possível. Nós da foodDelivery estamos muito felizes de termos você conosco nas últimas semanas. E já se foi um mês desde a sua admissão. Tá na hora de começarmos os trabalhos, hein? Ah, eu nem me apresentei, o meu nome é Larisse Cerqueira, mas pode me chamar de Lari, tá? Eu vou ser a sua buddy aqui na empresa. Eu estou enviando essa mensagem para lhe apresentar as suas primeiras atividades.

Nessa primeira atividade, o nosso objetivo é avaliar o seu conhecimento em análise de complexidade de soluções. Como você deve saber, o nosso core business está fundamentado na otimização da entrega de delivery de comida. Nós temos como objetivo oferecer o menor custo de entrega, no menor tempo possível para os consumidores e empresas parceiras. Para isso, nós contamos com vários parceiros entregadores e/ou outras empresas que fazem entregas como a gente. Nessa atividade, o nosso objetivo é te ambientar com o nosso roteirizador. Quando você despacha mais de um pedido para um motoboy entregar na mesma viagem, o nosso sistema planeja a melhor rota para que o entregador faça o caminho mais curto e não perca tempo. Mas calma que eu vou te explicar tudo mais detalhadamente.

O roteirizador, ou router-food, tem como objetivo planejar a melhor rota para o entregador fazer o caminho mais curto e não perder tempo na entrega. Porém, a pergunta chave deste problema é: como eu sei qual o caminho mais curto? Nós da food delivery entendemos como o caminho mais curto aquele em que o motoboy leve o menor tempo no trânsito entre o estabelecimento comercial (ponto a) e os seus destinos (pontos b, c, d e etc). Contudo, como eu defino o custo de tempo entre uma entrega e outra? Essa é a pergunta de milhões e você vai ter que responder. O seu objetivo é definir o custo de um ponto de entrega e outro, além da melhor rota a ser realizada pelo motoboy.

Vamos lá: o nosso router-food recebe como entrada um arquivo json com a configuração conforme o arquivo pedidos.js em anexo. Você deve ler o arquivo json que vai lhe informar o estabelecimento comercial e as entregas que precisam ser feitas pelo motoboy. Cada local de entrega deve ser considerado como um ponto a ser otimizado pelo seu router-food. Você

deve encontrar o menor caminho envolvendo todos os pontos de entrega dessa lista de entregas descrita no arquivo json. Obrigatoriamente, você deve fazer o motoboy sair do estabelecimento comercial e retornar para ele.

O nosso router-food funciona muito como uma heurística para o problema do Caixeiro Viajante. Nesse seu primeiro desafio, nós queremos que você mostre que entende essa classe de problema, assim como consegue codificar uma solução para ele. Então, eu vou definir algumas questões de projeto com você, tudo bem? Esse vai ser o nosso primeiro desafio.

Boa sorte!

### **Requisitos de projeto:**

**Linguagem de programação:** Java, JavaScript ou Python.

**Complexidade:** Análise de complexidade da sua solução algorítmica, apresentando os cálculos para encontrar a função de complexidade de tempo.

**Defina:** função de complexidade de tempo

Complexidade Assintótica de tempo em notação Big-O

**Resumo Técnico:** Até duas laudas, explicando o contexto do problema e a solução adotada.

### **Critérios de análise do seu código:**

Modularização

Legibilidade de Código

Documentação

### **Produto:**

Você deve implementar sua solução de heurística para o problema do Caixeiro Viajante, considerando como entrada de dados o arquivo json apresentado no Anexo I. Além da implementação de um algoritmo para o Caixeiro Viajante, você deve realizar a análise de complexidade da sua proposta de solução. Você deve definir a função de complexidade de tempo e a função de Complexidade Assintótica de tempo em notação Big-O. Você deve gerar um resumo técnico de até duas laudas, explicando o contexto do problema e a solução adotada por você. O seu projeto será pontuado da seguinte forma:

**Pontuação:**

| <u>Item</u>  | <u>Pontuação</u> |
|--|------------------|
| Implementação da solução em linguagem de programação | 10 pontos        |
| Análise de Complexidade                              | 8 pontos         |
| Resumo Técnico                                       | 2 pontos         |
| <b>Total</b>   | 20 pontos        |

**Prazos:**

| <u>Item</u>  | <u>Data</u> |
|--|-------------|
| Implementação da solução em linguagem de programação | 19/10/2023  |
| Análise de Complexidade<br>Resumo Técnico            | 26/10/2023  |