

# Projeto e Análise de Algoritmos

## Trabalho 2

Thiago Pinheiro de Araújo

EMAp/FGV

2023.1

## Objetivo

- Criar soluções para auxiliar a operação da UrbanFast, um novo serviço de entregas, com o objetivo de apoiar solicitações de clientes, parceiros e operadores do negócio.
- **Operação 1:** encontrar entregadores próximos.
- **Operação 2:** definir a rota de uma entrega simples.
- **Operação 3:** definir a rota de uma entrega considerando centros de distribuição.
- **Operação 4:** sugerir entregas adicionais com base em uma rota.

## Requisitos

- A UrbanFast irá iniciar a sua operação no Rio de Janeiro, utilizando uma versão digital da **planta da cidade** para definir rotas de entrega.
- Somente serão utilizadas **vias de mão dupla**, sem considerar o trânsito.
- Vias são compostas por **segmentos**, com extremidades conectando outros segmentos da mesma ou de outras vias
  - Cada segmento possui o seu tamanho em metros.
- Uma **rota** consiste em uma sequência de segmentos a serem percorridos
  - O tamanho de uma rota é definido pela soma do tamanho de cada segmento presente nela.

## Requisitos

- Um **vendedor** é um usuário que disponibiliza produtos para venda
  - Todo vendedor possui um catálogo de produtos para venda.
  - Cada produto possui um preço e um peso.
- Um **cliente** é um usuário que realiza a compra de um produto
  - Após a compra o cliente espera a entrega no seu endereço
    - Faz parte do trabalho definir a representação de um endereço.
- Um **entregador** é um usuário que recebe pedidos de entrega
  - Todo entregador possui um veículo, cujo tipo define o peso máximo de produtos que pode carregar.

## Requisitos

- Um **centro de distribuição (CD)** consiste em um armazém estrategicamente localizado para otimizar a entrega de produtos
  - Todo CD possui um inventário indicando os produtos armazenados e a quantidade disponível de cada um.
- Toda compra de um produto gera um **pedido**, que pode ser:
  - Simples, quando o entregador pega o produto na localização do vendedor e leva até a localização do cliente.
  - Otimizado, quando o entregador busca o produto em um CD e leva até a localização do cliente.
- Com base em um pedido é definida a rota de entrega.

- **Operação 1:** encontrar entregadores próximos.
- Quando um pedido for postado será necessário encontrar um entregador para fazer o transporte.
- Deve-se buscar os  $n$  entregadores mais próximos do local de coleta
  - Ou seja, os entregadores cujas rotas para o endereço do vendedor tenham o menor tamanho.
- A entrada da operação deverá ser o pedido, e o resultado a lista de entregadores sugeridos.

## Requisitos

- **Operação 2:** definir a rota de uma entrega simples.
- Caso o pedido seja simples, após receber a confirmação de um entregador precisamos encontrar a rota com a menor distância total para entrega.
- Essa rota deverá iniciar na posição corrente do entregador, considerar a coleta do produto no endereço do vendedor, e terminar no endereço do cliente.
- A entrada da operação deverá ser o pedido e o entregador escolhido, e o resultado a lista de segmentos representando a rota.

- **Operação 3:** definir a rota de uma entrega considerando centros de distribuição.
- Caso o pedido seja otimizado, um entregador deverá coletar o produto em algum centro de distribuição, e transportá-lo para o endereço do cliente.
- A operação consiste em buscar os  $n$  entregadores cujas rotas apresentem a menor distância total, considerando a passagem por algum centro de distribuição que contenha o produto no seu inventário.
- A entrada da operação deverá ser o pedido, e o resultado uma lista contendo em cada elemento: o entregador, o centro de distribuição, e a lista de segmentos representando a rota.



- **Operação 4:** sugerir entregas adicionais com base em uma rota.
- Buscando aproveitar o deslocamento de um entregador em uma rota, proponha uma solução para sugerir pedidos adicionais buscando maximizar o faturamento.
- Essa solução deverá considerar:
  - A capacidade de transporte do tipo do veículo.
  - O peso e o preço de cada produto.
  - A sua disponibilidade no inventário.

- A solução de software deverá ser composta por:
  - Um módulo implementando as quatro operações.
  - Uma aplicação para demonstrar a corretude através de cenários conhecidos
    - Ex: TDD ou BDD.
  - Uma aplicação para executar o experimento (descrita a seguir).
- Sugestão: crie as soluções em um ou mais módulos com interface(s) bem definida(s) e faça cada uma das aplicações utilizar esse(s) módulo(s).

- O programa deverá ser escrito em C++
  - Podem ser utilizadas bibliotecas para realizar operações de apoio.
  - Não podem ser utilizadas estruturas de dados ou algoritmos de bibliotecas para implementar as soluções específicas do trabalho.
- Todos os dados devem ser armazenados em memória utilizando as estruturas de dados implementadas durante o trabalho.
- Desconsidere o controle de acesso às operações e os tipos de usuário
  - O objetivo é desenvolver uma solução com base nos algoritmos abordados na disciplina.

## Dicas

- O código deve ser limpo e fácil de ler.
- Utilize uma nomenclatura clara e consistente ao longo do projeto.
- Escreva comentários em trechos de código para explicar ao leitor o comportamento idealizado.
- Procure fazer uma modelagem com encapsulamento adequado e baixo acoplamento.

## Experimento

- Instrumente as operações de forma que o tempo de execução seja medido a cada execução.
- Execute o programa exercitando todas as operações variando o número de entregadores, pedidos e CDs.
- Produza gráficos a partir dos resultados de medição de cada operação e compare com a complexidade esperada.

- O relatório do projeto deverá conter:
  - A modelagem arquitetural da solução.
  - Apresentação e descrição das estruturas de dados criadas.
  - A solução de cada um dos quatro problemas contendo:
    - O pseudo-código do algoritmo desenvolvido.
    - Uma discussão sobre a sua corretude.
    - A análise da sua complexidade.
    - Discussão dos resultados do experimento.
- O documento deverá ser entregue em PDF e deverá conter o nome dos integrantes do grupo.

- Faz parte do trabalho **interpretar o problema** e gerar uma modelagem adequada para as estruturas de dados e para os algoritmos.
- É esperado que essa modelagem seja influenciada por decisões de projeto tomadas com base no domínio da aplicação, desde que alinhadas com a especificação do trabalho.
- A modelagem e as decisões de projeto deverão ser justificadas no relatório.
- A avaliação de cada algoritmo irá considerar a complexidade (em tempo e espaço) atingida e a corretude de cada algoritmo
  - Procurem encontrar a solução com maior desempenho.

- A avaliação será baseada nos critérios a seguir:
  - Qualidade do relatório e discussão dos resultados.
  - Solução para encontrar entregadores próximos.
  - Solução para definir a rota de uma entrega simples.
  - Solução para definir a rota de uma entrega considerando centros de distribuição.
  - Solução para sugerir entregas adicionais com base em uma rota.
  - Estruturas de dados utilizadas.
  - Modelagem geral do programa.
  - Organização do projeto e qualidade do código.
  - Manual de instruções.



## Entrega

- Grupos devem ter de 4 a 5 integrantes.
- O trabalho deverá ser entregue em um arquivo zip contendo
  - O relatório.
  - O código-fonte da aplicação.
  - Um manual (readme) com instruções para compilar e executar.
- A entrega deverá ser realizada no eClass até o dia 08/12/2023 às 23:59
  - Trabalhos entregues após esta data perderão 1.0 ponto por dia.
  - Limite máximo: 11/12/2022 às 23:59.