

# Projeto e Análise de Algoritmos

Projeto

Thiago Pinheiro de Araújo

EMAp/FGV

2024.1

## Objetivo

- Criar soluções para auxiliar a prefeitura da cidade Vargas à planejar suas linhas de transporte, e fornecer um serviço de roteamento para transitar pela cidade de forma eficiente utilizando os serviços fornecidos.
- **Tarefa 1:** projetar as linhas de metrô da cidade.
- **Tarefa 2:** projetar a linha de ônibus *hop-on/hop-off* da cidade.
- **Tarefa 3:** criar um serviço para fornecer a rota mais rápida entre dois endereços utilizando os serviços da cidade.

## Requisitos

- A prefeitura da cidade Vargas irá fornecer a **planta da cidade** em uma versão digital.
- Uma **região** da cidade é composta por um conjunto de ruas
  - Cada região é identificada por um CEP (ex: 54312).
- Uma **rua** apresenta imóveis dispostos sequencialmente com uma numeração crescente.
- Um **imóvel** é uma edificação da cidade identificada unicamente pela tupla <CEP, Rua e número>
  - Pode ser residencial, comercial, industrial ou uma atração turística.

## Requisitos

- Um **segmento** é um conjunto de imóveis de uma rua separados por dois cruzamentos
  - Cada segmento apresenta um limite de velocidade para veículos e o seu tamanho em metros.
  - A faixa de veículos pode ser de sentido único ou duplo.
- Um **cruzamento** é formado pela conexão das extremidades de dois ou mais segmentos da mesma rua ou de ruas diferentes.
- Uma **pessoa** é um usuário que transita na cidade utilizando os meios de transporte disponíveis.

## Requisitos

- Os **meios de transporte** disponíveis em Vargas são:
  - Metrô.
  - Ônibus.
  - Taxi.
  - Deslocamento não-motorizado.
- Uma **rota** representa o caminho sugerido pelo sistema para uma pessoa transitar entre dois imóveis
  - É composta por uma sequência de segmentos a serem percorridos.
  - Pode combinar meios de transporte diferentes.

## Requisitos

- Um segmento pode ter sua velocidade máxima limitada pelo **trânsito** naquele instante
  - Ao computar uma rota entre dois endereços deve-se consultar uma API para indicar o trânsito corrente em um dado segmento.
- Estações metrô e pontos de ônibus deverão apresentar uma lista com os horários de chegada do veículo para embarque dos passageiros.

## Requisitos

- O **tempo** para percorrer um segmento é calculado com base no meio de transporte:
  - No meio **metrô** é calculado com base no tamanho do segmento e na velocidade do trem (70 km/h).
  - Nos meios **ônibus** e **taxi** é calculado com base em uma relação entre o comprimento do segmento, o limite de velocidade, e o trânsito no mesmo.
  - Nos meios **ônibus** e **metrô** deve-se considerar os horários de chegada do veículo no local de embarque.
  - No meio **deslocamento não-motorizado** deve-se considerar a velocidade média de um adulto andando a pé (5 km/h).

## Requisitos

- O **custo** para percorrer um segmento é calculado com base no meio de transporte:
  - Nos meios **metrô** e **ônibus** o custo é definido por uma taxa única para a rota. O custo de uma passagem metrô deverá ser sempre maior que o custo de uma passagem de ônibus.
  - No meio **taxi** o custo é calculado multiplicando a distância percorrida por uma taxa. Esse meio considera um valor mínimo por corrida, penalizando a sua utilização para trechos muito pequenos. Tanto a taxa quanto o valor mínimo deverão ser parâmetros configurados pela prefeitura da cidade.
  - No meio **deslocamento não-motorizado** não há custo.



## Requisitos

- **Tarefa 1:** projetar as linhas de metrô da cidade
  - Foi decidido que:
    - Todas as linhas de metrô serão escavadas embaixo das ruas da cidade.
    - Todas as estações serão construídas em um cruzamento.
    - Cada região possuirá uma estação de metrô.
  - A escolha de cada estação de metrô deverá buscar minimizar a distância entre ela e o ponto mais longe da sua respectiva região.
  - A planta da cidade indica para cada segmento um número representando o custo para escavá-lo, tornando-o parte da linha de metrô.

- **Tarefa 1:** projetar as linhas de metrô da cidade
  - Considere que a partir de uma estação de metrô uma pessoa poderá se locomover para qualquer outra estação
    - Existirão rotas para trens, no entanto não faz parte do trabalho defini-las.
  - Projete um algoritmo capaz de definir os segmentos a serem escavados, de forma que o custo para a cidade seja mínimo, no entanto todas as estações definidas sejam conectadas.

## Requisitos

- **Tarefa 2:** projetar a linha de ônibus *hop-on/hop-off* da cidade
  - A cidade terá uma única linha de ônibus, passando por todas as regiões da cidade.
  - A linha de ônibus deverá iniciar e terminar no mesmo lugar.
  - Deseja-se maximizar o número de imóveis comerciais e atrações turísticas no trajeto, e minimizar o número de imóveis residenciais e industriais.
  - Projete um algoritmo capaz de definir a rota para a linha de ônibus seguindo as considerações descritas acima.

## Requisitos

- **Tarefa 3:** criar um serviço para fornecer a rota mais rápida entre dois endereços utilizando os serviços da cidade
  - A cidade fornecerá uma aplicação móvel para auxiliar a mobilidade urbana.
  - Uma pessoa poderá informar endereços de origem e destino, e receber uma rota eficiente para locomoção (considerando partida imediata).
  - A rota deverá ser representada através de uma sequência de segmentos, que poderão ser percorridos utilizando meios de transporte diferentes.
  - A pessoa poderá informar o valor máximo que deseja gastar nessa locomoção.
  - Projete um algoritmo capaz de definir a rota que leve a pessoa em menos tempo entre a origem e o destino informados, considerando as demais características descritas acima.

## Implementação

- A solução de software deverá ser composta por:
  - Um módulo implementando as três tarefas.
  - Uma aplicação para demonstrar a corretude através de cenários conhecidos.
  - Uma aplicação para executar o experimento (descrito 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).
- A API de trânsito deve ser simulada através de uma função que retorna a velocidade máxima para um segmento passado como parâmetro
  - Implemente de forma que possa ser controlada globalmente pela aplicação do experimento.

- 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 cruzamentos e segmentos
  - Desative o trânsito para evitar que o mesmo tenha impacto nas medições.
- 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 problema contendo:
    - O pseudo-código do algoritmo projetado.
    - 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
  - Argumente sobre qualquer característica introduzida que influencie na otimalidade da solução.

- A avaliação será baseada nos critérios a seguir:
  - Qualidade do relatório e discussão dos resultados.
  - Solução para definir as linhas de metrô.
  - Solução para definir a rota de ônibus.
  - Solução para definir a rota mais rápida entre dois endereços.
  - Estruturas de dados utilizadas.
  - Modelagem geral do programa.
  - Organização do projeto e qualidade do código.
  - Manual de instruções para compilar e executar.

## 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 02/12/2024 às 23:59
  - Trabalhos entregues após esta data perderão 1.0 ponto por dia.
  - Limite máximo: 04/12/2024 às 23:59.