

# Relatório A1

Nome: Maykon Marcos Junior

Matrícula: 22102199

Elabore um relatório de uma página no formato PDF comentando para cada um dos exercícios quais as estruturas de dados selecionadas, justificando as escolhas. Não esqueça de informar os nomes dos integrantes da equipe.

## ***Criação dos grafos***

Struct Vértice: Possui um índice, um rótulo e um vetor com os índices dos vizinhos

Struct Arco: Possui os índices dos 2 vértices e o peso da ligação

vértices: um vetor de Vértice que armazena os vértices na ordem que foram lidos

arcos: um vetor de Arco que armazena todos os arcos na ordem que foram lidos

matriz: uma matriz de adjacências que faz o mesmo que o vetor arcos, mas permite consulta constante a um arco.

## ***Algoritmo Busca em Largura***

As estruturas para representar os vetores de distância (que não era necessário) e árvore de busca em largura foram os `std::vector`, o primeiro inicializado com MAX (uma constante = 2147483647) e o segundo como -1, já que armazena os índices ao invés do vértice em si. O vetor de já visitado também é um `std::vector` booleano, e a fila de visitas é uma `std::queue`. As escolhas basearam-se na eficiência e praticidade da biblioteca, e a razão por estruturas baseadas em vetor foi pelo acesso em tempo constante a suficiência para a representação necessária, economizando memória.

## ***Algoritmo Bellman-Ford***

Também foi usado `std::vector<int>` para representar a distância e a ancestralidade dos vértices (novamente, porque armazena o índice ao invés do vértice em si) de forma a simplificar a representação e facilitar a manipulação, economizando memória que seria gasta com uma estrutura dedicada.

## ***Algoritmo Floyd-Warshall***

Para esse algoritmo, foi usada uma matriz para as distâncias, que é um `std::vector<std::vector<int>>`, pelos motivos supracitados, o que permite rápida manipulação e fácil compreensão.

