**Relatório do Projeto**

|  |  |
| --- | --- |
| Nome do Integrante | TIA |
| Rodrigo Mattos Scavassin | 32055374 |

**Conteúdo do Relatório**

Descrição do problema

* Foi elaborado um grafo que representa a rede de transporte aéreo doméstica. Cada vértice representa um aeroporto e as arestas representam voos diretos entre dois aeroportos.
* Como há diferenciação entre ida e volta, o grafo foi definido como direcionado com peso na aresta sendo este a distância entre dois aeroportos em km e o tempo de viagem em minutos.
* O grafo em questão é considerado esparso.

Base de dados

* Base de dados obtida em: <https://sistemas.anac.gov.br/dadosabertos/Voos%20e%20opera%C3%A7%C3%B5es%20a%C3%A9reas/Dados%20Estat%C3%ADsticos%20do%20Transporte%20A%C3%A9reo/>
* Foram removidos os registros incompletos e com distância entre aeroportos igual a zero.
* Os dados obtidos mostram as viagens feitas desde 2000. Foram extraídos os dados referentes a todos os voos feitos em 2022.

Representação do grafo

Diagrama, Esquemático

Descrição gerada automaticamente

Mapa

Descrição gerada automaticamente com confiança média

Implementação

* Por ser considerado um grafo esparso, a melhor solução seria a de utilizar uma lista de adjacência. Porém o aluno optou em implementar o código utilizando uma matriz de adjacência.
* Total de vértices: 194
* Total de arestas: 1835
* a/v² ≅ 5%
* O código foi desenvolvido em C++

O código encontra-se na página do github:

<https://github.com/rodrigo-scavassin/grafos.git>

A apresentação encontra-se no youtube:

<https://www.youtube.com/watch?v=cNWYbrJD1KM>

Menu de opções

Texto

Descrição gerada automaticamente

Ler dados do arquivo

Lê o arquivo e mostra a mensagem:



Gravar dados no arquivo

Mostra a mensagem:



É possível conferir todas as alterações feitas.

Inserir vértice

Texto

Descrição gerada automaticamente

É possível conferir o vértice inserido quando abrimos o arquivo salvo.

Texto, Carta

Descrição gerada automaticamente

Inserir aresta

Texto

Descrição gerada automaticamente

É possível conferir a aresta inserida quando abrimos o arquivo salvo.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente com confiança média

Também é possível conferir a aresta inserida quando pedimos para imprimir o grafo



Caso a aresta já exista, a seguinte mensagem aparecerá:



Remove vértice

Mostra a mensagem



Número de vértices atualizado no arquivo

Texto

Descrição gerada automaticamente com confiança média

Número de arestas atualizado no arquivo

Uma imagem contendo Diagrama

Descrição gerada automaticamente

Não é possível encontrar o vértice removido no arquivo.

Remove aresta

Mostra a mensagem



Número de arestas atualizado no arquivo

Texto

Descrição gerada automaticamente

Se a aresta não existir, aparece a seguinte mensagem:



Mostrar conteúdo do arquivo

Texto

Descrição gerada automaticamente

...

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Mostrar Grafo

O grafo ficou muito grande como é possível verificar na captura de tela a seguir:

Padrão do plano de fundo

Descrição gerada automaticamente

Como o grafo é esparso, mas foi desenvolvido como matriz de adjacência, temos muitas arestas com o valor infinito.

Por este motivo, foi criado um método que apresenta apenas as arestas que contém um valor diferente de infinito.

Texto

Descrição gerada automaticamente

Mesmo assim, a visualização fica ruim por conta da quantidade de arestas muito alta.

**Etapa 2 do projeto**

Para a segunda etapa do projeto, foram desenvolvidos dois métodos principais. O primeiro método calcula a conexidade do grafo (C3, C2, C1, C0). O segundo método calcula a melhor rota entre dois aeroportos, tanto em termos de distância quanto tempo percorridos.

Para o cálculo da categoria de conexidade, foi utilizado o algoritmo de percurso em largura para determinar se o grafo é sfconexo. Então foram criados os métodos fconexo(), sfconexo() e desconexo(), todos retornando um valor booleano caso o grafo apresente esta conexidade, então o método determinaConexidade() imprime a categoria obtida.

No caso do projeto a categoria de conexidade obtida foi como mostra o resultado a seguir:



Para o cálculo de melhor rota, utilizou-se o algoritmo de Floyd para criar uma matriz de rotas, tempo e distância obtidas quando o usuário insere dois aeroportos.

Alguns exemplos obtidos:

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**Conclusão:** o projeto proposto foi desenvolvido com sucesso. Algumas sugestões de melhoria envolvem considerar o tempo de parada nos aeroportos (considerando peso nos vértices) e filtrar as rotas mais usuais.