Relatório - Uma aplicação de grafos dirigidos

O grafo consiste em uma estrutura abstrata que serve para modelar diversas situações e problemas reais. Um grafo G é composto pelo par (V, A), onde V é denominado conjunto de vértices e A é denominado conjunto de arcos. Um vértice $u \in V$ representa um objeto, enquanto um arco $(u, v) \in E$ representa uma relação de u para v. Um grafo é dito ponderado quando além da relação entre dois objetos existe um valor associado a cada arco. (ROSSI & MENA-CHALCO, 2018).

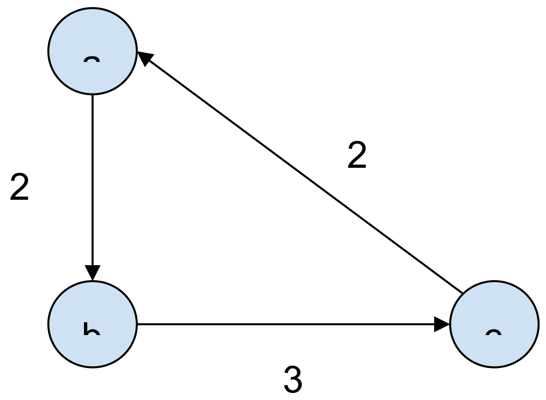
Grafos são utilizados para modelar situações de deslocamento. Podemos mapear cidades como vértices e estradas como arcos. E diversos são os problemas relacionados neste contexto. Imagine a situação onde um motorista deseja encontrar o menor caminho (ou o mais rápido) para chegar em seu destino. Podemos converter o mapa de uma cidade em um grafo e buscar um caminho (conjunto de arcos) mínimo (soma dos pesos dos arcos) que leve o motorista até seu destino.

O problema de caminho mínimo é um dos problemas genéricos intensamente estudados e utilizados em diversas áreas como Engenharia de Transportes, Pesquisa Operacional, Ciência da Computação e Inteligência Artificial. Isso decorre do seu potencial de aplicação a inúmeros problemas práticos que ocorrem em transportes, logística, redes de computadores e de telecomunicações (ATZINGEN et al., 2012).

Uma das variações do problema de caminho mínimo é o problema com de caminho mínimo de uma única origem. O algoritmo mais conhecido para esta variação é o algoritmo Dijkstra. Este algoritmo possui complexidade $O(|A| + |V| log|V|)$. O algoritmo consiste em um procedimento iterativo onde utiliza-se a estrutura de dados Fila de Prioridades para determinar o próximo vértice onde podemos determinar sua distância do vértice de origem (CORMEN et al., 2002).

Este trabalho teve como objetivo a utilização do algoritmo Dijkstra em um contexto real. No caso específico do trabalho, utilizamos o mesmo para encontrar o menor caminho de origem única em um mapa de dez cidades do Mato Grosso do Sul. Além disso, foi utilizado a ideia de potência de matrizes para geração de instâncias.

Inicialmente consideramos a matriz de adjacência a_{ij}, onde a_{ij} = *1* se $(i, j) \in A$, e a_{ij} = *0* caso contrário. Considere o seguinte grafo, sua matriz de adjacência e sua matriz distância correspondente.



Matriz Adjacência A

	a	**b**	**c**
a	0	1	0
b	0	0	1
c	1	0	0

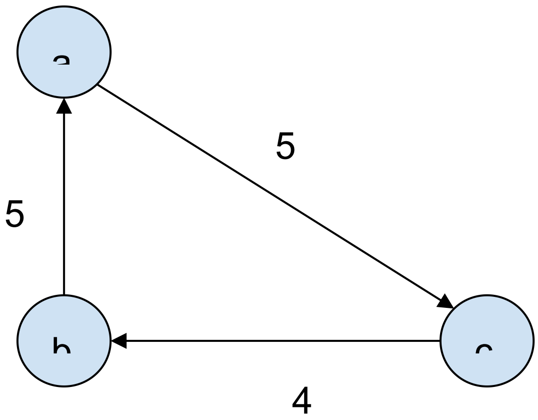
Matriz distância associada a A

	a	**b**	**c**
a	0	2	0
b	0	0	3
c	2	0	0

Uma operação entre matrizes consiste na exponenciação de matrizes. Mas qual o significado de A^2 dado que A é a matriz de adjacência. A^2 irá significar as adjacências a distâncias 2, ou seja, a_{ij}^2 armazena o número de caminhos de tamanho igual a dois arcos entre os vértices i e j.

O algoritmo de multiplicação de matrizes é apresentado no Algoritmo MultMatriz. Mas se utilizarmos esta mesma operação na matriz distância iremos contorcer este dado, já que o mesmo iria armazenar o produto ao invés da soma das arestas. O procedimento para calcular as distâncias é apresentado no Algoritmo

CalculaDistancia. Para exemplificar estes conceitos, é apresentado o grafo obtido pela potência dois do grafo apresentado anteriormente.



Matriz Adjacência A^2

	a	**b**	**c**
a	0	0	1
b	1	0	0
c	0	1	0

Matriz distância associada a A^2

	a	**b**	**c**
a	0	0	5
b	5	0	3
c	0	4	0

Podemos calcular a potência A^n de um grafo qualquer, desde que $n \leq |V|$. Neste trabalho geramos potências de dois possíveis do grafo original (A^2, A^4e A^8) e matrizes pesos (distâncias) associadas ($peso^2$, $peso^4$ e $peso^8$).

Para o cálculo de potências de matrizes A^n foi utilizado o método de exponenciação rápida de matrizes, descrito no Algoritmo ExpMatriz. Este algoritmo se utiliza do método de Divisão e Conquista (CORMEN et al., 2002) que permite calcular A^n com a complexidade de $O(n^3 log\, n)$ ao invés do algoritmo trivial $O(n^4)$. Esta complexidade pode ser ainda mais reduzida utilizando um algoritmo mais eficiente de multiplicação de matrizes, como é o caso do Algoritmo de Strassen (KRAUSE et al., 2016).

Geramos instâncias a partir das matrizes A e $peso$. Implementamos o algoritmo Dijkstra na Linguagem C++. Para validar a implementação proposta geramos 5 consultas e retornamos a distância mínima em cada uma delas. A tabela abaixo apresenta os resultados obtidos.

Matriz	Consulta 1	Consulta 2	Consulta 3	Consulta 4	Consulta 5
A^1	444	532	128	1066	71
A^2	444	532	1045	1066	1176
A^4	494	720	1045	1066	1318
A^8	778	1004	1375	1350	1318

Como já esperado, quanto maior a potência maior será o caminho mínimo, pois ao comprimirmos arcos iremos naturalmente aumentar os pesos dos mesmo. Abaixo são apresentados pseudocódigos citados durante o texto.

Algoritmo MultMatriz - Multiplicação de Matrizes

Entrada: Matrizes A (l x m) e B (m x n).
Saída: Matriz C (l x n), onde $C = A \cdot B$

Para $i \leftarrow 1, \ldots, l$ **faça**
  Para $j \leftarrow 1, \ldots, n$ **faça**
    $C_{ij} \leftarrow 0$
    Para $k \leftarrow 1, \ldots, m$ **faça**
      $C_{ij} \leftarrow C_{ij} + A_{ik} \cdot B_{kj}$

Retorne C

Algoritmo ExpMatriz - Exponenciação Rápida de Matrizes

Entrada: Matrizes A (n x n) e um inteiro positivo m.
Saída: Matriz B (n x n), onde $B = A^m$

Se m = *1* **então**

Retorne A

B ← ExpMatriz($\frac{m}{2}$)

Se m *% 2* = *0* **então**

Retorne MultMatriz(B, B)

Senão

Retorne MultMatriz(MultiMatriz(B, B), A)

Algoritmo CalculaDistancia - Calcula Distância

Entrada: Matrizes de distância $D1$ (l x m) e $D2$ (m x n) e matrizes de adjacência $A1$ (l x m) e $A2$ (m x n).

Saída: Matriz distância $D3$ (l x n), resultado do grafo resultante $A1 \cdot A2$

Para $i \leftarrow 1, \ldots, l$ **faça**

Para $j \leftarrow 1, \ldots, n$ **faça**

$D3_{ij} \leftarrow \infty$

Para $k \leftarrow 1, \ldots, m$ **faça**

Se $A1_{ik} \cdot A2_{kj} > 0$ **então**

$D3_{ij} \leftarrow min(D3_{ij}, D1_{ik} + D2_{kj})$

Se $D3_{ij} = \infty$ **então**

$D3_{ij} \leftarrow 0$

Retorne $D3$

Referências

ATZINGEN, J. V. et al. **Análise comparativa de algoritmos eficientes para o problema de caminho mínimo**. Universidade de São Paulo (USP). São Paulo. Escola Politécnica, 2012.

CORMEN, T. H. et al. Algoritmos: teoria e prática. **Editora Campus**, v. 2, p. 296, 2002.

KRAUSE, A. M. et al. Análise de Desempenho da Multiplicação de Matrizes por Strassen contra o Método Tradicional. **14th Workshop on Parallel and Distributed Processing**. 2016.

ROSSI, L.; MENA-CHALCO, J. P. Criação de grafos de tópicos do conhecimento baseada em genealogia acadêmica. **Encontro Brasileiro de Biblioimetria e Cientometria**, v. 6, 2018.