INSTITUTO DE INFORMÁTICA
Universidade Federal de Goiás

Algoritmo de Dijkstra e Detecção de Ciclos

Prof.Hebert Coelho da Silva
hebert@ufg.br

2023

INSTITUTO DE INFORMÁTICA
UFG

UFG
UNIVERSIDADE FEDERAL DE GOIÁS

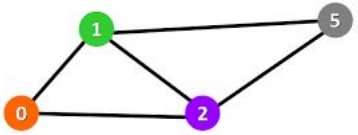
Alunos

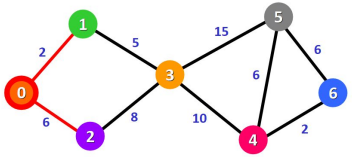
- JOAO VICTOR ROSA - 201905538
- KEVEN LUCAS VIEIRA GONDIM - 202000181
- RAFAEL SOUZA PORTO - 202003612

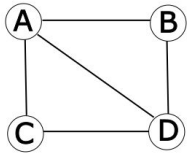
Introdução

Introdução Teoria dos grafos

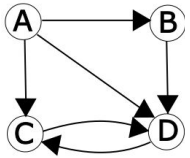
1. Um grafo é uma que representação abstrata de um conjunto de objetos e das relações existentes entre eles. É definido por um conjunto de nós ou vértices, e pelas ligações ou arestas, que ligam pares de nós.
2. Grafos dirigido: Aqueles onde as arestas possuem direção. Ou seja dois vértices podem ser vizinhos mas existir caminho apenas de um para o outro. Grafos não dirigidos, as arestas possuem os dois sentidos.
3. O grafo pode possuir arestas com peso ou sem, sendo que o peso é a medida que determina o custo ao se percorrer o grafo







Grafo A



Grafo B

Detecção de ciclos

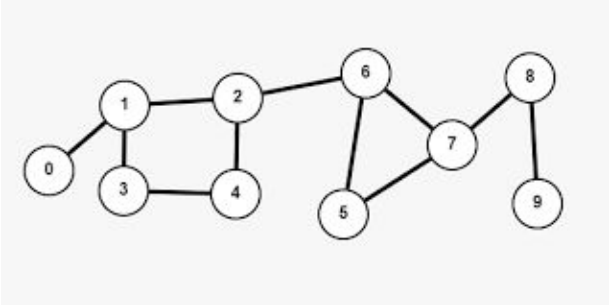
Layout: Main point

O que é Ciclo?

Um passeio e uma sequência de vértices, onde se v e w são consecutivos existe uma aresta que liga v a w

Um caminho é um passeio onde não existem repetições de arestas. É um caminho simples quando não tem vértices repetidos

Um ciclo é um caminho fechado, sou seja o primeiro e último vértice são os mesmos

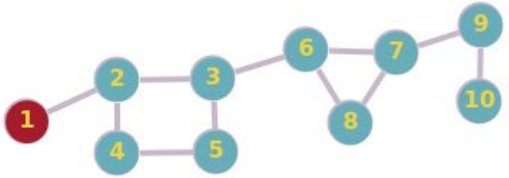


Para saber se um grafo tem ciclos o algoritmo de busca em profundidade(DFS) é aplicado em todos os vértices do Grafo.

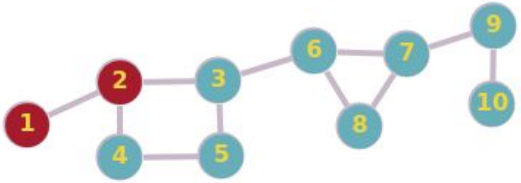
Para cada vértice visitado *v* se houver um vizinho já visitado que não é pai de *v* então existe ciclo no grafo.

Aplicação e execução

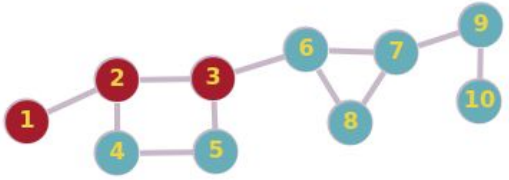
0



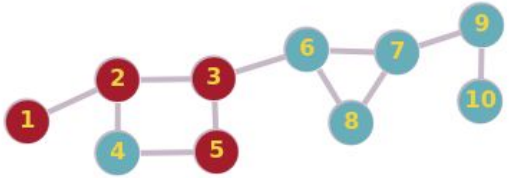
1



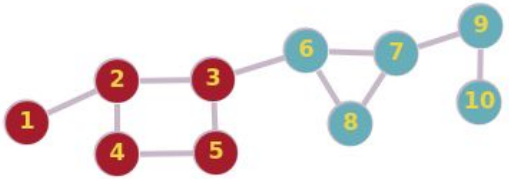
2

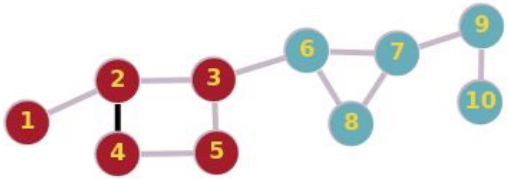


3



5





4 - Ciclo encontrado

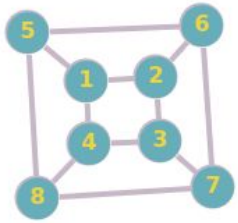
Obter Conjuntos dominante localizador aberto redundante em grafos cúbicos

Grafo cúbico: todos os vértices possuem 3 arestas

Conjunto dominante localizador aberto: Vértices que podem dominar seus vizinhos, e cada vértice do grafo é dominado por um subconjunto diferente.

É redundante quando um vértice pode falhar e as propriedades são mantidas.

Observação: Seja $S \subseteq V(G)$ um conjunto RED-OLD de um grafo cúbico G. Se um vértice $v \in V(G)$ é adjacente a um C_4, então temos $v \in S$.

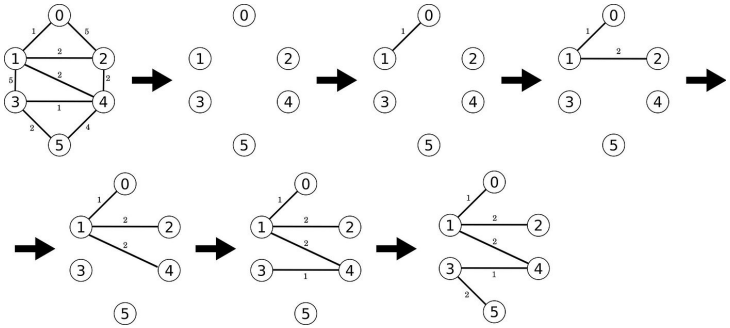


Fonte: S. J. Seo. Fault-tolerant detectors for distinguishing sets in cubic graphs. Discrete Applied Mathematics, 293:25–33, 2021.

Algoritmo de Dijkstra

Propósito e casos de uso

1. Identificar o caminho de menor custo entre o nó de origem e demais vértices em um grafo.
2. Gera uma árvore de caminho mínimo
3. Pode ser aplicado em GPS, para encontrar o melhor caminho para chegar a um lugar.
4. Todas as arestas do grafo devem possuir peso positivo para que o algoritmo funcione.



Noções Básica do algoritmo de dijkstra

- Distâncias Iniciais: Ele mantém um registro das distâncias mais curtas conhecidas de cada nó até o nó de origem, inicialmente com valores infinitos para todos os nós, exceto o nó de origem, cuja distância é definida como zero.
- Exploração: O algoritmo explora o grafo analisando as arestas e seus custos, buscando caminhos de menor custo.
- Atualização de Distâncias: Quando encontra um caminho com custo menor do que a distância atualmente conhecida para um nó, atualiza a distância do nó para refletir esse menor custo.
- Marcação e Adição ao Caminho: Após encontrar o caminho de menor custo para um nó, esse nó é marcado como "visitado" e adicionado ao caminho.
- Processo Repetido: O algoritmo continua iterando e repetindo o processo de exploração, atualização de distâncias e marcação até que todos os nós do grafo tenham sido visitado.

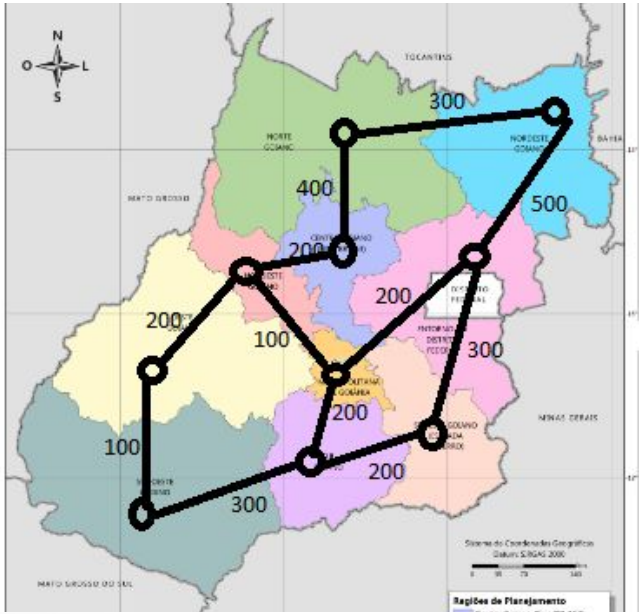
Aplicação e execução

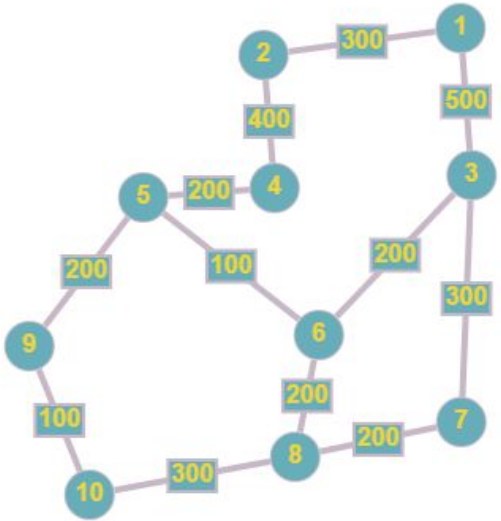
Modelagem

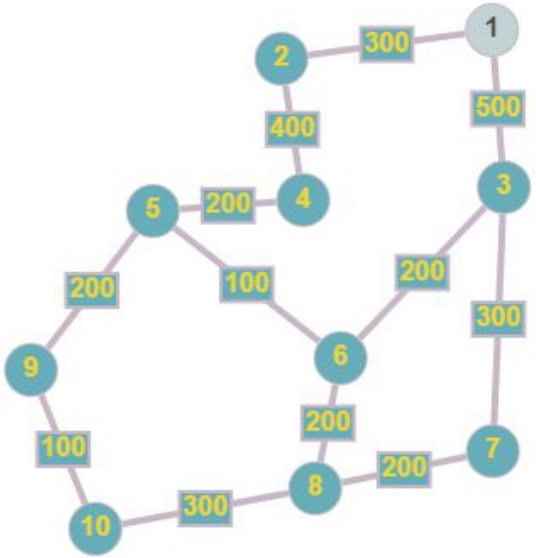
Uma determinada indústria, localizada no nordeste goiano, precisa descobrir a menor distância em quilometragem a ser percorrida até o seu centro de distribuição, localizado no sudoeste goiano, visando economizar os seus gastos com sua frota de caminhões, abaixo será fornecido uma tabela com as distâncias fictícias.

Região 1	Distancia	Região 2
Nordeste	300	Norte
Norte	400	Centro
Nordeste	500	Entorno DF
Centro	200	Noroeste
Noroeste	100	Metropolitana
Entorno DF	200	Metropolitana
Entorno DF	300	Sudeste
Sudeste	200	Sul
Metropolitana	200	Sul
Noroeste	200	Oeste
Sul	300	Sudoeste
Oeste	100	Sudoeste

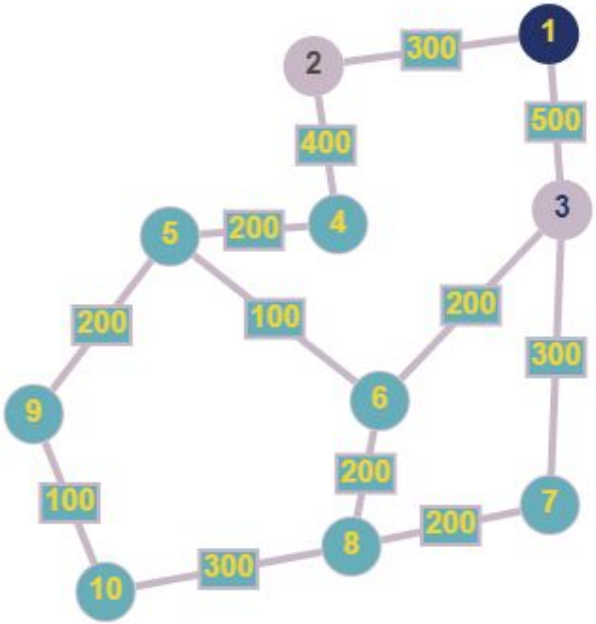
Modelagem


N
O
L
S
TOCANTINS
MATO GROSSO
MINAS GERAIS
300
400
500
200
200
200
100
300
200
100
200
300
Regiões de Planejamento

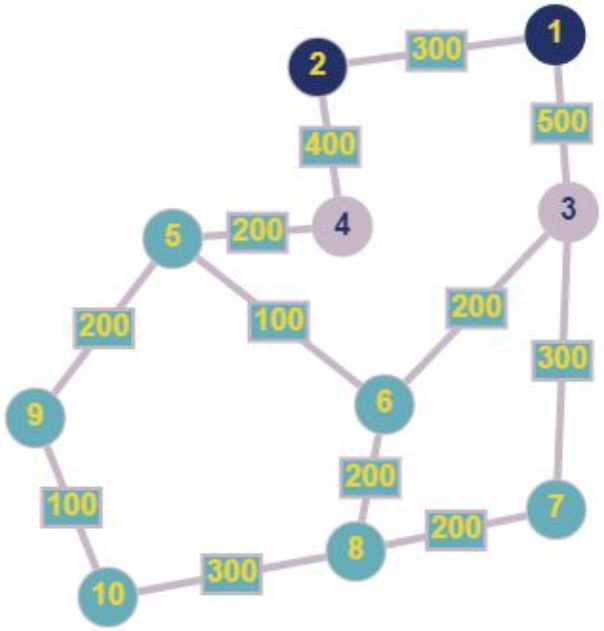

1
2
3
4
5
6
7
8
9
10
300
500
400
200
200
100
200
300
200
100
200
300



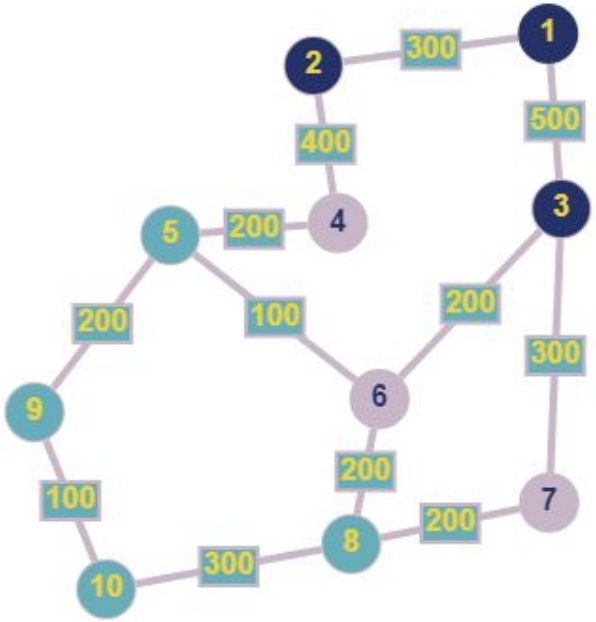
vértice	prédecessor	distância
1	NULL	0
2	NULL	∞
3	NULL	∞
4	NULL	∞
5	NULL	∞
6	NULL	∞
7	NULL	∞
8	NULL	∞
9	NULL	∞
10	NULL	∞



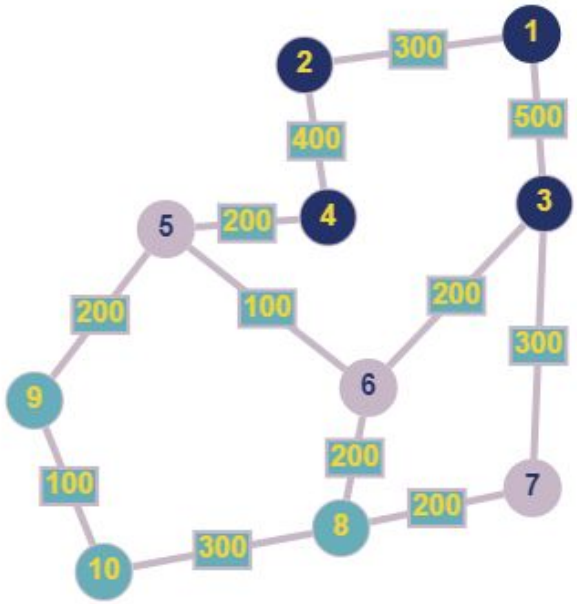
vértice	prédecessor	distância
1	NULL	0
2	1	300
3	1	500
4	NULL	∞
5	NULL	∞
6	NULL	∞
7	NULL	∞
8	NULL	∞
9	NULL	∞
10	NULL	∞



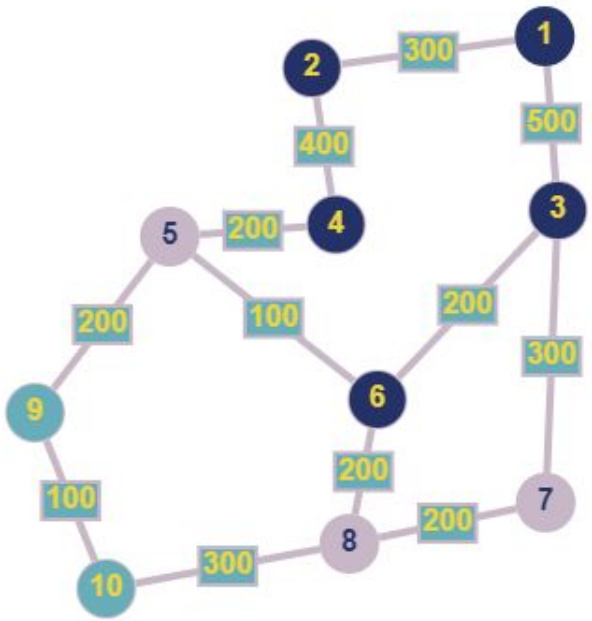
vértice	prédecessor	distância
1	NULL	0
2	1	300
3	1	500
4	2	700
5	NULL	∞
6	NULL	∞
7	NULL	∞
8	NULL	∞
9	NULL	∞
10	NULL	∞



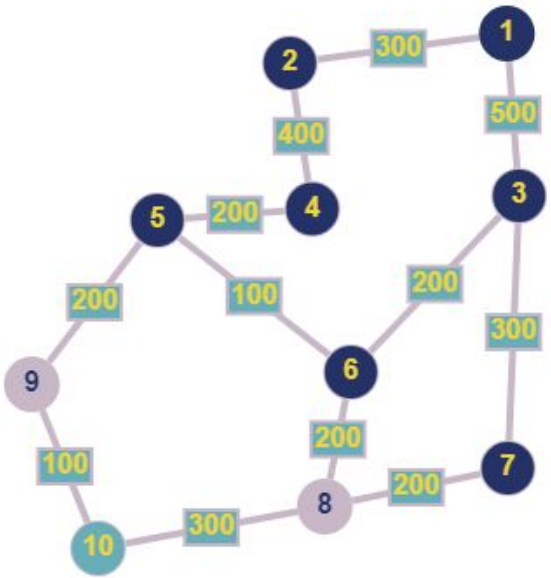
vértice	prédecessor	distância
1	NULL	0
2	1	300
3	1	500
4	2	700
5	NULL	∞
6	3	700
7	3	800
8	NULL	∞
9	NULL	∞
10	NULL	∞



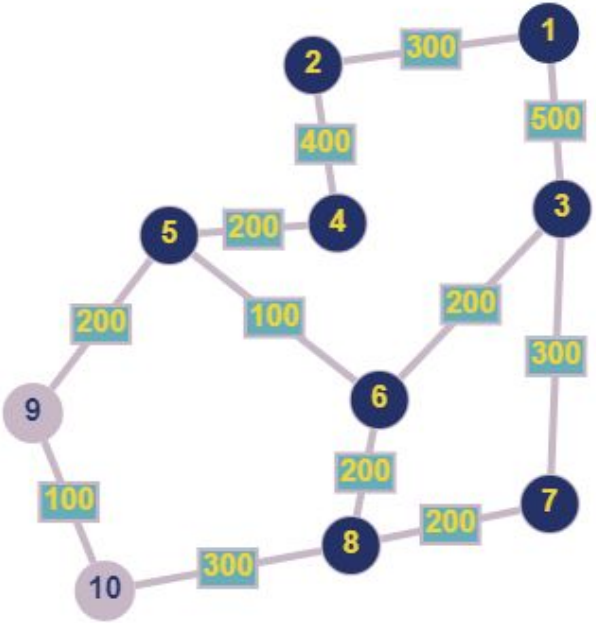
vértice	prédecessor	distância
1	NULL	0
2	1	300
3	1	500
4	2	700
5	4	900
6	3	700
7	3	800
8	NULL	∞
9	NULL	∞
10	NULL	∞



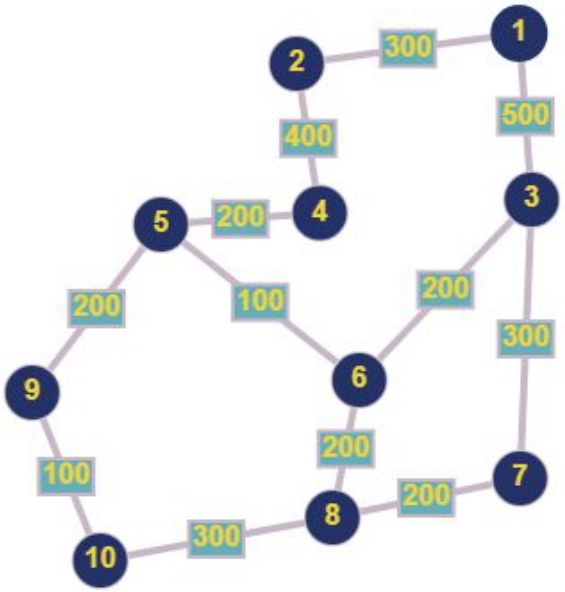
vértice	prédecessor	distância
1	NULL	0
2	1	300
3	1	500
4	2	700
5	6	800
6	3	700
7	3	800
8	6	900
9	NULL	∞
10	NULL	∞



vértice	prédecessor	distância
1	NULL	0
2	1	300
3	1	500
4	2	700
5	6	800
6	3	700
7	3	800
8	6	900
9	5	1000
10	NULL	∞



vértice	prédecessor	distância
1	NULL	0
2	1	300
3	1	500
4	2	700
5	6	800
6	3	700
7	3	800
8	6	900
9	5	1000
10	8	1200



vértice	prédecessor	distância
1	NULL	0
2	1	300
3	1	500
4	2	700
5	6	800
6	3	700
7	3	800
8	6	900
9	5	1000
10	9	1100

Referências

https://computerscience360.files.wordpress.com/2018/02/algoritmos-teoria-e-prc3a1tica-3ed-thomas-cormen.pdf