Algoritmo de melhor caminho com força bruta

Vitor Falcão Poncell

151082250

# Problema:

Qual a melhor rota entre dois dois nós de um grafo, problema bastante estudado pois com ele uma rota pode ser otimizada, seja ela uma rota de dados em uma rede, de um carro numa cidade, de entrega, entre outros.

# Solução:

O projeto foi desenvolvido com um algoritmo de força bruta para identificar a melhor rota possível.

No algoritmo de força bruta iremos encontrar todos os possíveis caminhos entre os nós inicial e final já calculando o custo que eles tem, somente após isso é que iremos verificar entre as possíveis soluções a com menor custo, como todas as possíveis soluções já foram mapeadas é possível coletar algumas informações além de a melhor solução, é possível listar todas as soluções ou todas as soluções que atendam um requisito de distância ou tempo máximo.

Algoritmo de força bruta não é o ideal para ser usado em um produto final, pois possui uma complexidade de exponencial, este projeto utiliza deste algoritmo para fins acadêmicos.

# Projeto:

Projeto foi feito na IDE Eclipse, pode ser acessado em:

https://github.com/VitorPoncell/Graphs.

Com finalidade de demonstração, foi criado um gerador de grafos que cria aleatoriamente um grafo dado uma quantidade de nós, ou o grafo pode ser lido de um arquivo de texto onde se tem uma conexão por linha no formato “nó,distância,tempo,nó”.

Em caso de grafos gerados O grafo gerado pode ser escrito num arquivo em um formato utilizado pelo graphviz, assim podemos ter uma representação gráfica do grafo, podemos ter essa representação gráfica do grafo passando o código gerado do grafo no site: https://dreampuf.github.io/GraphvizOnline/

vale salientar que é verificado no código se um nó já foi visitado antes na possível solução a ser calculada desconsiderando ele, para assim evitar que a busca entre em loop.

private void calc (String from, String to, String semiResult, int semiDistance, int semiTime) {

for (Path path : mGraph.node(from).getPaths().toList()) {

if (path.getTo().equals(to)) {

mResuts.add(new Result(semiResult + "-" + path.getTo(), semiDistance + path.getDistance(), semiTime + path.getTime()));

continue;

}

if (semiResult.contains(path.getTo())) {

continue;

}

calc(path.getTo(), to, semiResult + "-" + path.getTo(),semiDistance + path.getDistance(),semiTime + path.getTime());

}

}

# Teste:

Dado um grafo gerado randomicamente quantas interações e em quanto tempo ele irá executar em 100 testes em média, para chegar no resultado do melhor caminho

Tabela número de nós x iterações x tempo

|  |  |  |
| --- | --- | --- |
| Número de nós | Interações médias | Tempo (ms) |
| 8 | 199 | 0,68 |
| 10 | 2387 | 1,34 |
| 12 | 46702 | 9,52 |
| 15 | 2606988 | 223,4 |

# Conclusão:

Força bruta não o melhor jeito de se encontrar um melhor caminho, podemos perceber que tempo gasto cresce exponencialmente ao aumentar o número de nós, porém um ponto positivo é que ao se calcular uma vez todos os caminhos são mapeados a informação o melhor caminho é somente um filtro de menor tempo ou distância entre as possíveis soluções, por exemplo, ao se calcular o melhor caminho do nó 1 ao nó 10 será calculado todos os caminhos possíveis então somente depois será aplicado um filtro para retornar o caminho com mentor distância, se for perguntado por exemplo os caminhos possíveis com distância menor que 20 não é preciso calcular novamente, basta aplicar um filtro onde a distância total dos caminhos encontrados tem de ser menor ou igual a 20.