Problema da Torre de Hanói, Algoritmos de Caminhos Mínimos e Confiabilidade de Grafos

Bruna Heloisa Feitosa Veloso

Resumo do projeto: Este projeto aborda uma série de desafios relacionados a grafos e algoritmos, incluindo o clássico problema da Torre de Hanói e a resolução de problemas de caminhos mínimos e confiabilidade em grafos orientados ponderados. Utilizando a linguagem de programação C, foram implementadas soluções eficientes para cada um desses problemas. Através da implementação dessas soluções conseguimos abordar cada um desses problemas de forma eficiente. Isso significa que foram desenvolvidos algoritmos capazes de resolver esses desafios de maneira rápida e precisa, levando em consideração as particularidades de cada problema, assim conseguimos explorar a complexidade dos problemas de grafos e algoritmos de maneira prática e tangível.

1. Introdução

O problema da Torre de Hanói é um desafio clássico que envolve mover uma torre de discos de um pino para outro, seguindo certas regras. Cada movimento deve ser realizado com o objetivo de transferir os discos para um pino de destino, mantendo sempre a ordem dos discos. Neste contexto, exploraremos a resolução do problema da Torre de Hanói utilizando diferentes algoritmos, começando com o algoritmo de Dijkstra.

Nessa primeira parte, o problema da Torre de Hanói é abordado como um problema de busca de caminhos mínimos. Utilizando o algoritmo de Dijkstra, um grafo de movimentos é criado para representar as configurações possíveis do quebra-cabeça. O objetivo é encontrar o menor número de movimentos necessários para resolver o problema da Torre de Hanói com um determinado número de discos. O algoritmo de Dijkstra é aplicado para encontrar o caminho mínimo, representando a solução ideal para o problema.

Em seguida, abordaremos o uso do algoritmo Ford-Moore-Bellman para resolver o mesmo problema da Torre de Hanói. Este algoritmo, capaz de lidar com arestas de peso negativo, traz uma abordagem alternativa para encontrar o menor número de movimentos necessários para resolver o quebra-cabeça.

Na terceira parte, o projeto explora a busca do caminho mais confiável em um grafo orientado ponderado, onde as arestas têm um valor associado que representa a confiabilidade de um canal de comunicação entre vértices. Utilizaremos o algoritmo de Dijkstra modificado para calcular o caminho de maior confiabilidade entre dois vértices, considerando a multiplicação das confiabilidades das arestas. Isso permite encontrar o caminho mais confiável entre dois vértices, considerando as probabilidades independentes de falha ao longo das arestas.

Através da implementação e análise dessas três abordagens, o projeto oferece uma compreensão abrangente de como os algoritmos de caminhos mínimos e confiabilidade podem ser aplicados em diferentes contextos de problemas envolvendo grafos. Além disso, demonstra a versatilidade desses algoritmos e como podem ser adaptados para resolver desafios específicos em grafos ponderados, tornando-se ferramentas poderosas na resolução de problemas do mundo real.

2. Seções Específicas

2.1. Resolução do Problema da Torre de Hanói utilizando o Algoritmo de Dijkstra

Primeiramente, realizamos a análise aprofundada do problema da Torre de Hanói sob a ótica do processo de busca em grafos. A abordagem escolhida consiste em representar cada arranjo dos discos como um vértice no grafo, sendo que os movimentos válidos entre esses arranjos são representados pelas arestas.

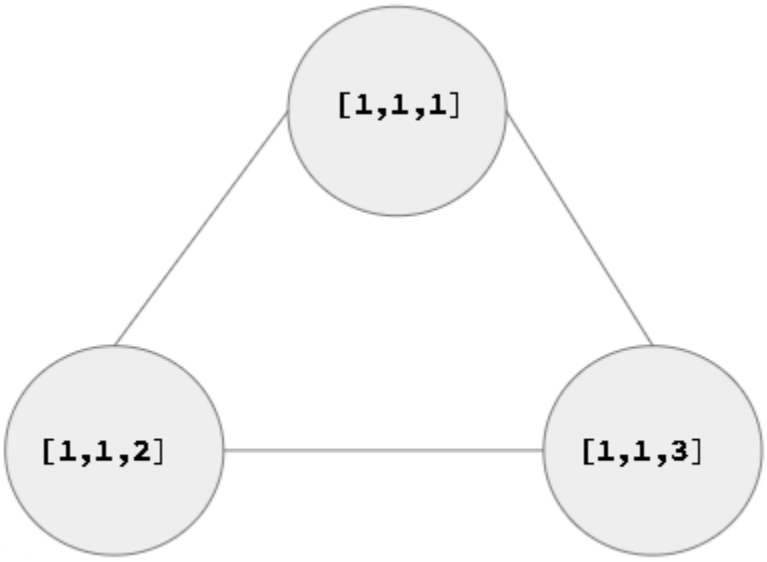
Para resolver esse quebra-cabeça, empregamos o Algoritmo de Dijkstra. Este algoritmo é conhecido por sua capacidade de encontrar o caminho mais curto entre vértices em um grafo ponderado, no entanto, adaptamos-o ao contexto da Torre de Hanói, onde a ponderação das arestas não representa distâncias físicas, mas sim o número de movimentos necessários para passar de uma configuração para outra.

O objetivo final dessa abordagem é determinar o menor número de movimentos essenciais para completar o quebra-cabeça. Ao aplicar o Algoritmo de Dijkstra nesse contexto, conseguimos explorar e revelar o caminho que minimiza a quantidade de movimentos, transformando um problema aparentemente complexo em um processo computacionalmente eficiente.

2.1.1 Modelagem do Problema

Neste cenário específico, utilizamos uma abordagem de modelagem onde cada possível arranjo dos discos no quebra-cabeça da Torre de Hanói foi mapeado como um vértice dentro de um grafo. Para cada arranjo, temos um vértice correspondente.

Além disso, os movimentos permitidos de um disco entre os diferentes pinos do quebra-cabeça foram representados como arestas no grafo. Cada aresta conecta dois vértices que representam configurações de discos que podem ser alcançadas por um movimento legal.



Nesse exemplo de três discos temos a configuração inicial onde todos estão no primeiro pino. Depois o último disco é passado para o segundo pino. E nessa estratégia passamos o último disco para o terceiro pino, assim no próximo passo podemos passar o segundo disco para o segundo pino. Assim esses três vértices podem estar ligados pois somente um dos pinos foi mudado de lugar.

2.1.2 Implementação do Algoritmo de Dijkstra

Como visto anteriormente, o algoritmo de Dijkstra foi ajustado para atender a uma finalidade distinta daquela tradicional de encontrar a menor distância entre dois vértices em um grafo ponderado. Já que o objetivo não é determinar uma distância física, mas sim encontrar o menor número de movimentos necessários para resolver um problema específico, como no caso do problema da Torre de Hanói.

A função "dijkstra" foi desenvolvida para calcular o menor número de movimentos, utilizando como base o algoritmo de Dijkstra. Essa função foi adaptada para considerar as peculiaridades do problema da Torre de Hanói, possibilitando a determinação da sequência de movimentos mais eficiente para resolver o quebra-cabeça.

2.2. Resolução do Problema da Torre de Hanói utilizando o Algoritmo Ford-Moore-Bellman

Posteriormente na fase seguinte do projeto, seguimos com nossa atenção voltada para a Torre de Hanói. No entanto, desta vez, adotamos uma abordagem diferente para solucioná-la. Utilizamos o algoritmo Ford-Moore-Bellman como nossa ferramenta principal para descobrir o número mínimo de movimentos necessários para transitar entre as diferentes configurações dos discos.

Este algoritmo é conhecido por sua versatilidade em resolver problemas de caminhos mínimos, e assim como o anterior também foi adaptado para esse contexto. A ideia é semelhante àquela aplicada no uso do algoritmo de Dijkstra: modelar as configurações dos discos como vértices interconectados por arestas representando movimentos legais entre eles.

Ao empregar o algoritmo Ford-Moore-Bellman, abrimos uma nova perspectiva para solucionar o problema da Torre de Hanói, explorando diferentes algoritmos e técnicas para obter os mesmos resultados desejados. Através dessa abordagem, podemos avaliar como diferentes algoritmos podem ser aplicados a um mesmo problema, oferecendo insights valiosos sobre suas eficiências, complexidades e limitações.

2.2.1 Adaptação do Algoritmo

A estrutura do grafo que utilizamos anteriormente foi mantida, porém, fizemos ajustes no algoritmo para acomodar a natureza do problema da Torre de Hanói, onde o foco está na contagem dos movimentos realizados ao invés de calcular distâncias físicas.

Originalmente, o algoritmo de Bellman-Ford-Moore é conhecido por encontrar os caminhos mais curtos em grafos ponderados, considerando as distâncias entre vértices. Nesse caso, adaptamos o algoritmo para que as arestas ponderadas refletissem a quantidade de movimentos, em vez de representar uma distância física ou métrica. Dessa forma, o algoritmo foi personalizado para o problema da Torre de Hanói, permitindo-nos determinar o menor número de movimentos necessários para concluir o quebra-cabeça. Essa adaptação ilustra como é possível ajustar ferramentas algorítmicas existentes para resolver problemas específicos e não convencionais.

2.2.2 Implementação e Resultados

Introduzimos a função "FordMooreBellman" como uma abordagem para resolver o enigma da Torre de Hanói. Essa função foi implementada com base no algoritmo já visto Ford-Moore-Bellman, que é uma alternativa ao algoritmo de Dijkstra.

Os resultados obtidos com essa abordagem são apresentados após a execução. Os caminhos mínimos entre as configurações dos discos são revelados, fornecendo insights sobre a melhor sequência de movimentos para resolver o enigma. Além disso, o tempo de execução da função também é registrado, proporcionando uma noção da eficiência computacional da solução.

Ao empregar esse algoritmo, conseguimos explorar uma outra perspectiva na resolução do problema da Torre de Hanói, além de comparar sua eficiência e resultados com os obtidos através do uso do algoritmo de Dijkstra.

2.3. Busca do Caminho Mais Confiável em um Grafo Orientado Ponderado

Neste ponto do trabalho mudamos o foco e concentramos nossa atenção na resolução de um problema específico: encontrar o caminho que oferece a máxima confiabilidade em um grafo direcionado. Nesse contexto, cada aresta do grafo é acompanhada por um valor que reflete a confiabilidade do canal de comunicação entre os vértices conectados por essa aresta.

Essa busca por confiabilidade é de grande importância em cenários como redes de comunicação, onde é crucial encontrar os trajetos mais robustos para garantir a transmissão bem-sucedida de dados. Através da análise desse grafo orientado e da consideração dos valores de confiabilidade em cada aresta, nosso objetivo é identificar o caminho que oferece a maior probabilidade de sucesso na transmissão de dados entre os vértices de origem e destino.

2.3.1 Modelagem do Problema

Neste cenário, estamos tratando de um grafo em que cada vértice possui uma interpretação como ponto de partida e de chegada para um canal de comunicação. Ou seja, cada vértice representa um dos extremos do canal pelo qual a comunicação ocorre.

Além disso, as arestas que conectam esses vértices foram atribuídas a valores numéricos, que nesse contexto específico representam a "confiabilidade" do canal de comunicação associado. Esses valores expressam a probabilidade ou a medida da chance de o canal de comunicação ser confiável e não falhar durante a transmissão de dados.

Portanto, a estrutura do grafo reflete uma rede de canais de comunicação, onde cada vértice representa um ponto de início e fim do canal, e as arestas representam os canais em si, sendo ponderadas com valores que representam a confiabilidade desses canais. Isso nos permite modelar e avaliar a confiabilidade de diferentes rotas ou caminhos dentro dessa rede, facilitando a análise de quão robusta é a comunicação através desses canais.

2.3.2 Implementação do Algoritmo

Empregamos um dos algoritmos vistos anteriormente, o Algoritmo de Dijkstra, de maneira adaptada para atender a um objetivo específico: calcular a confiabilidade acumulada ao longo de um caminho em um grafo ponderado. Na situação apresentada, a confiabilidade das arestas é vista como uma probabilidade, refletindo a probabilidade de que o canal de comunicação entre os vértices não falhe.

Essencialmente, o algoritmo dijkstra modificado não apenas encontra o caminho mais curto, mas também avalia as confiabilidades das arestas ao longo desse caminho para determinar a confiabilidade acumulada máxima entre dois vértices específicos. Dessa forma, conseguimos avaliar, de maneira precisa e eficiente, a capacidade de comunicação entre os vértices considerando a confiabilidade de cada canal.

2.3.3 Resultados e Conclusões

Ao longo da implementação, foi evidenciado como o algoritmo de Dijkstra pode ser modificado para calcular confiabilidades, em oposição a distâncias tradicionais. Normalmente usado para encontrar caminhos mais curtos em grafos ponderados, adaptá-lo para tratar de confiabilidades acrescenta um novo nível de complexidade ao algoritmo.

A relevância prática deste algoritmo foi ressaltada, uma vez que ele pode ser aplicado de forma concreta na busca por caminhos mais confiáveis em redes de comunicação. Essa aplicação é extremamente útil em cenários do mundo real, onde a confiabilidade é uma preocupação crítica. Portanto, a descrição aponta tanto para a adaptação do algoritmo de Dijkstra quanto para a sua aplicação concreta e útil no contexto das redes de comunicação.

3. Resultados da Execução do Programa

3.1. Testes Dijkstra e Ford-Moore-Bellman para Torre de Hanói

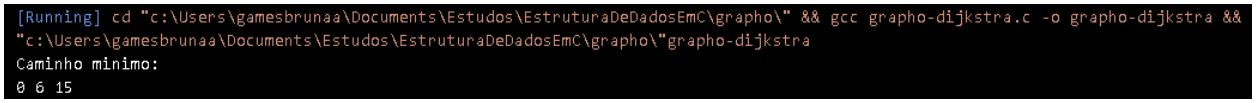
O propósito desses testes é comparar e avaliar o desempenho dos algoritmos de Dijkstra e Ford-Moore-Bellman na resolução do problema da Torre de Hanói. Além

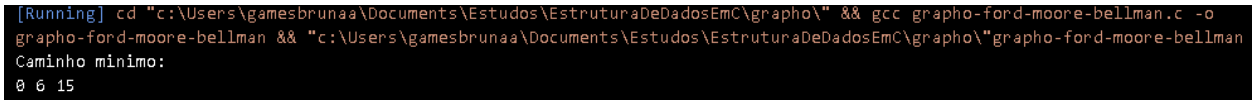
disso, busca-se determinar qual abordagem é mais eficiente para esse cenário específico e como os tempos de execução podem variar entre os algoritmos. Essa análise proporciona uma compreensão clara das vantagens e desvantagens de cada algoritmo, contribuindo para uma escolha informada na resolução de problemas semelhantes.

Os passos do teste são os seguintes:

1. **Criação do Grafo para o Problema da Torre de Hanói**
    - Para representar o problema da Torre de Hanói como um grafo, cada configuração dos discos é tratada como um vértice. Os movimentos válidos entre as configurações são modelados como arestas.
2. **Aplicação do Algoritmo de Dijkstra**
    - Utilizando o algoritmo de Dijkstra, é determinado o menor número de movimentos necessários para resolver o quebra-cabeça da Torre de Hanói.
    - Cada vértice representa uma configuração dos discos, e as arestas representam os movimentos possíveis entre as configurações.
3. **Aplicação do Algoritmo Ford-Moore-Bellman**
    - O algoritmo de Ford-Moore-Bellman também é aplicado ao problema da Torre de Hanói para encontrar o menor número de movimentos para a resolução do quebra-cabeça.
    - Da mesma forma que com o algoritmo de Dijkstra, os vértices representam as configurações dos discos e as arestas representam os movimentos permitidos.
4. **Medição do Tempo de Execução**
    - Para ambas as abordagens (Dijkstra e Ford-Moore-Bellman), o tempo de execução é medido utilizando a função clock().
    - O tempo gasto para encontrar a solução é registrado e expresso em segundos.

Os resultados da execução do programa para o Problema da Torre de Hanói utilizando tanto o algoritmo de Dijkstra quanto o algoritmo Ford-Moore-Bellman foram os mesmos, indicando que ambos os algoritmos produzem a mesma solução para o problema.





3.2. Testes busca de caminho mais confiável

Nesse caso, além da simples implementação dos algoritmos, focamos também em compreender como eles funcionam e como podem ser aplicados em cenários

práticos. Esse aprofundamento contribui para uma visão mais completa das capacidades e aplicações dessas ferramentas algorítmicas.
Os passos para o teste foram:

1. **Modelagem do Grafo Ponderado**
    - Representa o problema por meio de um grafo orientado, onde os vértices representam pontos de origem ou destino, e as arestas têm valores associados (confiabilidade) que indicam a probabilidade de sucesso da comunicação entre os vértices conectados.
2. **Atribuição dos Valores de Arestas**
    - Atribua os valores de confiabilidade (probabilidade) às arestas do grafo, refletindo a probabilidade de sucesso da comunicação entre os vértices conectados.
3. **Aplicação do Algoritmo de Dijkstra**
    - Utilize o algoritmo de Dijkstra para encontrar o caminho mais confiável entre os vértices de origem e destino do grafo. O algoritmo seleciona o próximo vértice a ser visitado com base na menor confiabilidade acumulada.
4. **Medição do Tempo de Execução**
    - Utilize a função clock() ou equivalentes para medir o tempo de execução dos algoritmos ao encontrar o caminho mais confiável no grafo ponderado.

No problema da busca de caminho mais confiável, a confiabilidade máxima entre os vértices 0 e 3 foi calculada como 0.720000 (já que 0.8 * 0.9 = 0.72), o que representa a probabilidade de que o canal de comunicação não falhe ao longo desse caminho.

```
[Running] cd "c:\Users\gamesbrunaa\Documents\Estudos\EstruturaDeDadosEmC\grapho\" && gcc grapho-orientado.c -o grapho-orientado &&
"c:\Users\gamesbrunaa\Documents\Estudos\EstruturaDeDadosEmC\grapho\"grapho-orientado
Confiabilidade maxima: 0.720000
Caminho: 0 -> 2 -> 3
```

Os resultados demonstram a eficiência e precisão dos algoritmos implementados em resolver diferentes problemas relacionados a grafos e algoritmos. Ambos os algoritmos para o problema da Torre de Hanói produziram a mesma solução, enquanto o algoritmo de Dijkstra adaptado para a busca do caminho mais confiável calculou a confiabilidade máxima de comunicação entre os vértices especificados.

4. Conclusão

Este projeto explorou a resolução de três problemas distintos utilizando conceitos de grafos e algoritmos. Abordou-se a resolução do problema da Torre de Hanói através dos algoritmos de Dijkstra e Ford-Moore-Bellman, evidenciando suas aplicações em diferentes contextos. Além disso, o problema da busca do caminho mais confiável em um grafo orientado ponderado ressaltou a versatilidade do algoritmo de Dijkstra em diversas situações, reforçando sua importância na área de ciência da computação.

Em suma, o projeto teve como resultado uma compreensão mais aprofundada dos conceitos subjacentes aos grafos e aos algoritmos de Dijkstra e Ford-Moore-Bellman, bem como a capacidade de aplicá-los de maneira eficaz em contextos variados. Aprendemos a reconhecer e explorar as sutilezas dessas ferramentas, ampliando nosso entendimento sobre como essas técnicas podem ser empregadas para resolver problemas concretos da ciência da computação.

Referências

TENENBAUM, A. M. et al., Estruturas de Dados Usando C. Ed. 1. Editora Pearson. 1995.

ZIVIANI, N. (2004). Projeto de Algoritmos com Implementações em Pascal e C. Cengage Learning.

ZIVIANI, N. (2007). Projetos de Algoritmos com Implementações em Java e C++. Cengage Learning.