PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

CIÊNCIA DA COMPUTAÇÃO

TRABALHO 1

Teoria de Grafos e Computabilidade

Gabriel Cunha Schlegel

Kaiky França da Silva

Samuel Horta de Faria

Belo Horizonte

Março de 2024

Sumário

1 Introdução

Este trabalho apresenta uma análise comparativa entre dois algoritmos para detecção de ciclos em um grafo: **Busca em Profundidade** (DFS) e **Permutação**. Ambos foram implementados utilizando listas de adjacência **(vector<vector<int»)** para representar o grafo, já que possui uma maior eficiência em termos de tempo de acesso e uso de memória, comparada a representação porr meio de matriz de adjacência.

2 Metodologia

2.1 Busca em Profundidade (DFS)

O algoritmo de DFS percorre o grafo explorando caminhos e armazenando os vértices visitados. Caso retorne a um vértice já visitado e o ciclo tenha pelo menos três nós, ele é identificado. Para evitar contagem duplicada, foi utilizado um conjunto **(set<vector<int»)** para armazenar apenas ciclos únicos.

2.2 Permutação

O método de permutação gera todas as possíveis sequências de vértices e verifica quais delas formam ciclos válidos. Embora garanta a identificação de todos os ciclos, esse método possui um custo computacional **extremamente alto**.

3 Análise Comparativa

A principal diferença entre os métodos está no desempenho conforme o tamanho do grafo aumenta:

- **DFS**: mais rápido e eficiente para grafos pequenos, mas pode consumir muita memória devido às chamadas recursivas em grafos grandes.
- **Permutação**: extremamente preciso, porém inviável para grafos grandes, pois seu tempo de execução cresce de forma significativa.

4 Conclusão

O algoritmo de **DFS** demonstrou ser a melhor opção para detecção de ciclos, pois mantém um bom equilíbrio entre eficiência e uso de memória. O método de permutação, apesar de garantir a exatidão na identificação de ciclos, é inviável para grafos grandes devido ao seu crescimento fatorial.

A escolha da lista de adjacência como estrutura de representação do grafo proporcionou um melhor desempenho, evitando custos excessivos de armazenamento.