Pontifícia Universidade Católica de Minas Gerais (PUC Minas)
ICEI - Instituto de Ciência Exatas e Informática
Engenharia de Computação
Algoritmos em Grafos

Trabalho Prático N°2

Alunos(as): Ítalo G. Carvalho, Leonardo Augusto de S. Filho

Professor: Zenilton Kleber Gonçalves do Patrocínio Júnior

Data: Maio de 2021

1. Introdução

No contexto do mundo atual os Algoritmos baseados em grafos vêm se tornado cada vez mais presentes na resolução de diversos problemas, relacionando as propriedades dos grafos a aplicações na nossa realidade. Uma das propriedades dos grafos que tem diversas aplicações é a de ciclo, que pode ser definido como um passeio entre 3 no mínimo três vertices sendo que o primeiro e o último vértice coincidem, mas nenhum outro vértice é repetido.

Neste contexto foi proposto na disciplina de Algoritmo em Grafos o problema de enumerar todos os ciclos de um grafo, que tem aplicações na área de topologia de redes, para solucionar este problema duas abordagens foram propostas: a primeira baseada na permutação dos vértices do grafo; a segunda a baseada em caminhamento no grafo.

2. Permutação

Primeiramente foi necessário encontrar todas as permutações das arestas do grafo, partindo de permutações com um tamanho de 3 arestas até n, onde n é a quantidade de vértices do grafo, para isso foi aplicado um algoritmo de heap que gera cada permutação, temos que quanto mais vértices o grafo tiver mais permutações serão geradas em todos os tamanhos, que torna extremamente ineficiente tal abordagem especialmente em grafos com muitos vértices. A medida que o heap encontra uma permutação a mesma é enviada para uma função que busca esta permutação na lista de adjacência do grafo carregado, comparando cada aresta da permutação com as arestas do grafo, caso está permutação esteja no grafo ela é mostrada na tela. Neste código além da grande quantidade de permutações cada uma delas deve ser comparada com a lista de adjacência, que tem seu tamanho associado a quantidade de arestas. Com alguns teste obtemos:

Quantidade de Vértices	Quantidade de Arestas	Tempo
4	6	117 ms
5	10	691 ms
6	15	2560 ms
7	21	19,396 s
8	28	213,796 s
9	36	44,86 min

Tabela 1. Resultados das medições de tempo dos testes realizados na permutação de vértices.

Analisando os dados conclui-se que este tipo de implementação é na grande maioria dos casos inviável, pelo seu alto custo computacional que até mesmo em pequenos grafos se mostra extremamente ineficiente. Aumentar tanto a quantidade de vértices quanto a de arestas aumentam gradativamente o custo computacional. Aumentando os vértices a quantidade de permutações também aumenta, já aumentar a quantidade de arestas influência na quantidade de comparações na consulta à lista de adjacência do grafo, e aumentar ambos em conjunto faz com que a execução utilize cada vez mais recursos computacionais.

3. Caminhamento no grafo

Como segunda implementação e como comparativo ao método de **permutação de vértices**, o método de enumerar ciclos por caminhamento no grafo se trata de uma busca em profundidade(DFS - Depht-first search) com uma lista para a verificação da existência de ciclos através dos vertices.

O algoritimo de busca em profundidade parte de um vertice do grafo(vertice de origem) adicionado-o ao inicio de uma lista. Apartir desse vertice, se aprofunda nos vertices vizinhos(vertices ligados ao vertice origem), assim a busca em profundidade e executada no novo vertice encontrado que é adicionado ao final da lista assim sucetivamente na ordem que os vertices vão sendo descobertos. Considerando-se que o vértice adicionado se trata de um vertice repetido e diferente do vertice de origem, este é removido do final da lista. Se um vertice encontrado for igual ao vertice de origem, significa que existe uma aresta de retorno caracterizando um ciclo e em seguida e impresso na tela e um contador é atualizado.

Quantidade de Vértices	Quantidade de Arestas	Tempo
4	6	16 ms
5	10	71 ms
6	15	129 ms
7	21	190 ms
8	28	259 ms
9	36	334 ms

Tabela 2. Resultados das medições de tempo dos testes realizados utilizando a busca em profundidade.

A Tabela 2 contém os dados dos testes de execução do algoritimo de 4 a 9 vertices, com suas devidas quantidades de arestas, e o tempo de execuçao da função responsável pela busca dos ciclos. E assim pode-se afirmar que o tempo de execução com o aumentar do numero de vértices é gradativo e estavél, tendo uma variação de aproximadamente 50ms a cada vértice adicionado.

4. Análise

Para analisar e comparar o desempenho de cada método, foram utilizados grafos completamente interconectados, ou seja, cada vértice era ligado aos outros vértices. Com isso espera-se que ambos os métodos sejam forçados ao seu máximo.

Com os resultados obtidos nos experimentos em ambas as abordagens, observou-se uma grande diferença entre elas. A abordagem da permutação dos vértices se mostra muito ineficiente, especialmente quando comparada a abordagem do caminhamento no grafo que gerou resultados mais limpos e com um custo computacional mais baixo.

Tamanha diferença nos resultados se deve ao fato da permutação analisar muitas combinações de vértices desnecessárias diversas vezes e em cada permutação um novo loop é feito para se comparar com a lista de adjacência, tornando o algoritmo extremamente custoso. Já no caminhamento no grafo é utilizada a busca em profundidade que gera uma lista dos vértices visitados, eliminando a possibilidade de se analisar arestas que não existem no grafo, o meio de se buscar os ciclos é através da localização de arestas de retorno que não gera um loop de busca.