Trabalho 1 Algoritmos e Grafos 2021.1
Encontrar um circuito Euleriano

André Uziel de Souza Dantas - 119051475

Um caminho euleriano consiste em um caminho que visita todas as arestas do grafo exatamente uma vez. Já o circuito euleriano, que é o tema deste trabalho, consiste em um caminho euleriano que começa e termina no mesmo vértice.

Para esse trabalho, usei o algoritmo de fleury, que consiste em:

1. Verificar se o grafo tem todos os vértices de grau par (circuito euleriano) ou exatamente 2 vértices de grau ímpar (caminho euleriano)
2. Comece por qualquer vértice do grafo no primeiro caso e por um dos vértices de grau ímpar no segundo.
3. Para cada aresta ligada ao vértice, escolha sempre a que **não** é de corte, a não ser que não haja nenhuma. Nesse caso, escolha alguma de corte.
4. Exclua a aresta pela qual você passou e o vértice se ele ficar isolado
5. Repita as etapas 3-4 até que o vértice atual não tenha mais arestas

A minha implementação consiste em 3 classes: AlgGrafos (principal), Graph e Vertex. Quando executado, o programa procura pelo arquivo grafo01.txt na pasta myfiles e interpreta ele conforme a convenção estabelecida. Como o enunciado pressupõe um grafo não direcionado, se um vértice está na lista de adjacências de outro, a recíproca também será verdadeira, então mesmo que isso não esteja explícito no arquivo, o programa adicionará os vértices na vizinhança um do outro.

Após a leitura do arquivo, será criada uma cópia do grafo, a fim de não modificar o original (pensando em algum uso futuro), a partir daí todas as operações serão feitas nessa cópia. Em seguida, serão removidos todos os vertices de grau zero, ou seja, os vértices que não interferem em nada no circuito euleriano. Se após essa remoção o grafo não for conexo, não existirá circuito euleriano, pois alguma aresta nunca será alcançada. Se o grafo apresentar algum vértice de grau ímpar, também não há circuito euleriano, pois essa é sua condição de existência. Nesses casos, será impressa uma mensagem na tela.

Após essas verificações, finalmente podemos iniciar a parte iterativa do algoritmo. Por padrão, o primeiro vértice do circuito é o de id 1. Em seguida, iteramos por todas as arestas desse vértice, a fim de achar uma ideal (que não seja de corte) para seguirmos. Essa decisão é feita removendo essa

aresta e verificando se o grafo continua conexo: se ainda for, achamos uma aresta própria e partimos para o próximo vértice; se não for, restauramos essa aresta e fazemos o mesmo para a próxima.

Para verificar se o grafo é conexo, foi criada a função is_connected, que se resume a uma busca em largura (BFS) seguido de uma verificação se todos os vértices foram alcançados por ela. Se foram, o grafo é conexo, caso contrário o grafo foi partido em 2 com a remoção da aresta e deixou de ser conexo. Uma vez removida a aresta correta, adicionamos o vértice ao qual ela leva ao circuito euleriano e repetimos o processo a esse vértice, removendo o anterior se ele ficar isolado. Perceba que quando salvamos o vértice na lista encadeada do circuito euleriano na realidade estamos adicionando o vértice correspondente no grafo original; Isso foi pensando em usos futuros.

Fontes:

[Fleury's Algorithm for printing Eulerian Path or Circuit - GeeksforGeeks]
Algoritmos Teoria e Prática - Cormen