



UNIVERSIDADE FEDERAL DE SANTA CATARINA
INE5413 - GRAFOS

RELATÓRIO DA ATIVIDADE 3

BRENDA SILVA MACHADO

2023.1

1. Código Fonte

A linguagem de programação escolhida para a realização desta atividade foi Python, voltada para a Orientação Objetos. Foi necessário o uso da biblioteca “Collections”. O código fonte da atividade está disponível no GitHub, no link:

<https://github.com/Brenda-Machado/grafos>

2. Fluxo Máximo

A primeira questão pede a implementação do algoritmo de Edmonds-Karp para encontrar o fluxo máximo de um grafo.

Assim como em trabalhos anteriores, adotei a estrutura de classe e de métodos para implementar. As variáveis seguiram a estrutura do grafo para facilitar a implementação e não necessitar de mudanças na estrutura de indexação e mapeamento, listas de tuplas no caso de arcos e dicionários no caso de vértices e funções. S e T permaneceram como inteiros, mantendo a estrutura de vértices. Fluxo máximo também é um inteiro, uma vez que é apenas um valor a ser somado. Por fim, a rede residual é um grafo que sofre a adição dos arcos residuais.

Os caminhos, a fila e os visitados estão na forma de lista, para preservar a ordem em que são inseridos de forma simples e fácil de acessar a indexação, assim como o método de inserir.

3. Emparelhamento Máximo

A segunda questão pede a implementação do algoritmo de Hopcroft-Karp para obter o emparelhamento máximo de um grafo bipartido. Para isso foi criada uma classe de grafo bipartido, sem pesos e arcos.

As variáveis do tipo x e y foram implementadas na forma de set(), para manter uma ordenação nos valores. Já mate e distâncias estão na forma de dicionários por causa da estrutura de key, que permite armazenar um valor para determinado vértice, o que facilita na hora de fazer as comparações e modificar os valores.

O diferencial é a estrutura deque q no BFS, consiste numa fila ordenada de duas extremidades, ou seja, é possível adicionar no começo e no fim. Além disso não há restrição na manipulação dos itens, o que foi crucial para a escolha dessa estrutura, além do método de remoção dela, como o popleft()

4. Coloração Mínima

A terceira questão pedia a implementação de um algoritmo que retornasse a coloração mínima de um grafo não-dirigido e não-ponderado. O algoritmo escolhido foi o algoritmo de Lawler.

A estrutura do algoritmo segue a estrutura de uma classe. A variável x ficou na forma de uma lista de tamanho $2^{|V|}-1$, de forma semelhante o S e a variável auxiliar `index_dict` (criada em decorrência de problemas na indexação das variáveis), mas com tamanho $2^{|V|}$.

As demais variáveis utilizadas também seguiram o padrão de lista, para manter a simplicidade do código e a indexação ficar padronizada.