

Relatório Atividade 3 de Grafos (INE5413)

Paulo B. F. Almeida, Pedro H. A. Silva, Wesly C. Ataíde

28 de novembro 2019

O seguinte relatório foi pedido como questão 4 da atividade 3. O trabalho utilizou, em boa parte, da classe “*Graph*” desenvolvida na atividade 1 e portanto continuamos utilizando a linguagem *python*. Sendo assim, as estruturas de dados já presentes na linguagem foram utilizadas na implementação dos algoritmos. A seguir são comentados cada exercício e as estruturas de dados utilizadas.

No exercício 1, pediu-se uma implementação do algoritmo de Edmonds-Karp. Usamos um dicionário para armazenar os arcos da rede residual, do mesmo jeito que armazenamos os arcos do grafo original. Na busca em largura do Edmonds-Karp para procurar caminhos aumentantes, usamos uma lista para armazenar o caminho.

O exercício 2 consistiu em implementar o algoritmo de Hopcroft-Karp. Mantivemos a bipartição como duas listas X e Y e rodamos o algoritmo. Na busca em largura e na busca em profundidade, tomamos a posição zero do vetor como a correspondente a posição nula. Os emparelhamentos correspondentes se encontram no vetor mate, que percorremos para mostrar as arestas.

No exercício 3 pediu-se um algoritmo que resolva a coloração de vértices, retornando o número mínimo de cores necessárias e o número cromático para cada vértice. Representamos os vários conjuntos e subconjuntos com listas. Para endereçar os subconjuntos nas listas que são usadas, criamos uma função que mapeia cada subconjunto a uma posição. Para obtermos os subconjuntos independentes maximais do grafo, procedemos a força bruta, retornando-os armazenados numa lista. Para obter as cores de cada vértice, mantivemos mais uma lista (*cor_set*), com os subconjuntos independentes maximais (codificados para a posição no vértice), que era atualizada junto com o número de cores necessárias. Obtemos dos conjuntos os vértices correspondentes de cada cor.