

Universidade Federal de Santa Catarina
CTC - Centro Tecnológico
INE - Departamento de Informática e Estatística.
INE5413 - Grafos
Aluno 1: Cainã Correa Caldas
Aluno 2: Vinicius Guedes dos Santos

Atividade 3

Questão 1 - Edmonds-Karp) Afim de resolver esse problema, primeiramente criamos um algoritmo capaz de converter um arquivo de entrada de um grafo de fluxo para um arquivo formatado como um grafo típico (aquele utilizado nas demais atividades). Uma vez convertido o arquivo para o formato que podíamos ler, foi preciso criar as estruturas de entrada para o algoritmo, isto é, o grafo de fluxo e a rede residual. A rede residual foi criada fazendo uma cópia do grafo original e em seguida adicionando-se um arco de retorno com peso zero para cada arco existente no grafo original. Criado o grafo e a rede residual, bastou invocar nossa implementação do algoritmo de Edmonds-Karp passando como parâmetro tais objetos. Para executar o algoritmo em uma dada entrada, basta alterar a variável caminho ao final do código para o caminho do arquivo com a instância de teste.

Questão 2 - Para esse exercício, implementamos um grafo utilizando lista de adjacências, que é vantajoso na hora de fazer buscas pelos vizinhos de um vértice. Além disso, passamos o algoritmo para a linguagem Python. Esbarramos, entretanto, em dificuldade na hora de definir na linha 13 da BFS o Dmatey. Todo o mate foi definido como null na linha 2 do Hopcroft-Karp, portanto como acessaríamos um vetor indexado em null? Superamos essa dificuldade através de definir as funções getD e setD, que levavam em consideração a possibilidade de a chave ser null, e escolhendo a posição zero para guardar o valor.

Questão 3 - Lawler) Não implementamos esse algoritmo.