



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA - INE  
CURSO DE CIÊNCIAS DA COMPUTAÇÃO  
PROFESSOR RAFAEL DE SANTIAGO

RELATÓRIO DO TRABALHO III DE GRAFOS - INE5413

João Paulo Araujo Bonomo (21100133)  
Gabriel Lima Jacinto (21202111)  
Rodrigo Santos de Carvalho (21100139)

Florianópolis  
Junho, 2023

# 1 Implementações

## 1.1 Edmonds-Karp

Para o desenvolvimento do nosso algoritmo ***Edmonds-Karp*** nos baseamos nos **Algoritmos 23 e 24** da apostila da disciplina. Usamos da estrutura de *std::vector* do C++ para representar nossa *Matriz de Adjacencia* e a *Rede Residual*. Na busca em largura (*BFS*), utilizamos de uma fila de inteiros, como implementedado no **Algoritmo 24** da apostila, usando a estrutura de *std::queue* do C++. Além disso, usamos de outras estruturas mais conhecidas do C++, como *arrays* e ponteiros.

## 1.2 Hopcroft-Karp

Nosso algoritmo de ***Hopcroft-Karp*** foi baseado nos **Algoritmos 27, 28, 29 e 30** da apostila da disciplina, e também conta com uma método auxiliar para printar a saída como esperado pelo enunciado do trabalho, chamada *printHopcroftKarp*. Para essa implementação, usamos da construção de um grafo bipartido para calcularmos propriamente o valor de cada conjunto de vértices. Quanto às estruturas de dados utilizamos de *std::vectors*, *arrays* e *lists*.

## 1.3 Coloração de Vértices

Usamos um algortimo guloso para implementar o nosso próprio ***Algoritmo de Coloração de Vértices*** em C++. Assim como os demais algoritmos e na própria representação do **Grafo** em si, usamos bastante o *std::vector*, *lists* e *pointers*. Ao final da execução, temos uma função auxiliar para imprimir o resultado chamada *printColoring*.