



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

RELATÓRIO DO TRABALHO II DE GRAFOS - INE5413

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

Florianópolis
maio, 2023

0.1 Componentes Fortemente Conexas

Para o desenvolvimento de nosso algoritmo de ***Componentes Fortemente Conexas*** nos baseamos nos **Algoritmos 15 e 16** da apostila da disciplina. Usamos da estrutura de *std::stack* do C++ para representar nossa *Stack*. Além disso, usamos de outras estruturas mais conhecidas do C++, como *lists* e ponteiros.

0.2 Ordenação Topológica

Nossa ***Ordenação Topológica*** foi baseada nos **Algoritmos 15 e 16** da apostila da disciplina, e também conta com uma método auxiliar para printar a saída como esperado pelo enunciado do trabalho, chamada *printTopologicalSorting*. Para essa implementação, usamos de *std::vectors*, *shared pointers* e estruturas mais complexas, como um vetor de *shared pointers*. O método *topologicalSorting()*, da classe *DirectedGraph*, reflete o **Algoritmo 15**, enquanto o método *DFSVisitTopologicalSorting()* reflete o **Algoritmo 16**.

0.3 Algoritmo de Prim

Usamos o **Algoritmo 22** da apostila para implementar o nosso próprio ***Algoritmo de Prim*** em C++, com o adicional de que usamos uma estrutura de dados *Heap* para encontrar as prioridades em **tempo logarítmico**, melhorando significativamente a eficiência computacional de nosso algoritmo. Assim como os demais algoritmos e na própria representação do **Grafo** em si, usamos bastante o *std::vector*, *lists* e *pointers*.