Relatório A2

Nome: Maykon Marcos Junior
Matrícula: 22102199

Componentes Fortemente Conexas

As estruturas usadas foram do tipo std::vector<int>, para permitir o acesso randômico constante e também o retorno de vários vectores (usando std::vector<std::vector<int>>) na função DFS (a necessidade de vários vetores int é para que estejam na mesma estrutura).

Ordenação Topológica

Foi usado std::vector<bool>, para o vetor de conhecidos, porém foi feito o uso de std::deque<int> para a ordenação. A escolha se deu devido à combinação de acesso randômico constante (amortizado) e a operação de inserir no início (evitando a necessidade de revertê-la depois).

Kruskal

Usou um vetor de Arcos (struct própria da classe) para a árvore geradora mínima final e, para as árvores intermediárias (que são conjuntos no pseudo-código), foi usada a estrutura discutida em sala de aula, a cd-Elemento:

```
- struct cd_Elemento
- {
-    cd_Elemento *pai;
-    int rank;
-    cd_Elemento()
-    {
-       pai = this;
-       rank = 0;
-    }
-
-    void ligar(cd_Elemento *x, cd_Elemento *y)
-    {
-       if (x->rank > y->rank)
-       {
-          y->pai = x;
-       }
-       else
-       {
-          x->pai = y;
-          if (x->rank == y->rank)
```

```
-                {
-                    y->rank++;
-                }
-            }
-        }
-
-        cd_Elemento *encontrar(cd_Elemento *x)
-        {
-            if (x->pai != x)
-            {
-                x->pai = encontrar(x->pai);
-            }
-            return x->pai;
-        }
-        void uniao(cd_Elemento *x, cd_Elemento *y)
-        {
-            ligar(encontrar(x), encontrar(y));
-        }
-    };
-
```