

**Atividade A2 – Grafos (INE5413)**  
**Ciências da Computação – Universidade Federal de Santa Catarina**  
**Prof. Rafael de Santiago**

Observações gerais:

- Trabalho deve ser executado em no máximo 3 estudantes da disciplina.
- Todas as codificações devem estar em uma das seguintes linguagens de programação: C/C++, Python ou Java.
- A biblioteca de grafos criada na Atividade 1 deverá ser utilizada na codificação dos demais itens dessa atividade.
- A entrega do código-fonte deverá ser realizada no MOODLE<sup>a</sup> em um arquivo compactado no formato ZIP ou TAR.GZ.
- A entrega do relatório é obrigatória. O relatório deve estar no formato PDF.
- A atividade vale 12 pts. As equipes que atingirem mais de 10 pts no trabalho, receberão nota 10 e o saldo será utilizado no próximo trabalho com nota inferior a 10 no semestre corrente.
- Duas ou mais equipes com trabalhos total ou parcialmente iguais receberão nota 0.

<sup>a</sup>A entrega deve ser realizada através do ambiente da turma no MOODLE.

1. **[Componentes Fortemente Conexas]** (3,0pts) Crie um programa que receba um grafo dirigido e não-ponderado como argumento. Ao final, imprima na tela as componentes fortemente conexas desse grafo. O exemplo abaixo trata de uma saída válida, na qual identificou-se duas componentes fortemente conexas  $\{3, 4, 5\}$  e  $\{1, 2, 6, 7\}$ .

```
3,4,5
1,2,6,7
```

2. **[Ordenação Topológica]** (3,0pts) Crie um programa que receba um arquivo de grafo dirigido não-ponderado com vértices rotulados como argumento. O programa deve fazer executar uma Ordenação Topológica. Depois exiba a ordem topológica, utilizando os rótulos de cada vértice, como no exemplo abaixo:

```
Acordar → DesligarDespertador → CalçarSandalias → LevantarDaCama → TomarBanho → EscovarOsDentes → PrepararCafe → PrepararOvosMexidos → TomarCafeDaManha → LavarLouças → EscovarOsDentes2 → CalçarMeias → VestirUniforme → ColocarSapato → FecharCasa.
```

3. **[Kruskal ou Prim]** (3,0pts) Crie um programa que recebe um grafo não-dirigido e ponderado como argumento. Ao final, o programa deverá determinar qual a árvore geradora mínima. O programa deverá imprimir o somatório das arestas na árvore na primeira linha e as arestas que pertencem a árvore geradora mínima na segunda linha, como no exemplo abaixo:

```
22
5-4, 0-1, 2-3, 3-0, 4-1
```

4. **[Relatório]** (3,0pts) Elabore um relatório de uma página comentando para cada um dos exercícios quais as estruturas de dados selecionadas, justificando as escolhas. Não esqueça de informar os nomes dos integrantes da equipe.

## Padrão de Arquivo de Entrada

O arquivo de entrada deve estar no formato abaixo. Na primeira linha,  $n$  é o número de vértices. Nas linhas seguintes e antes da palavra “\*edges”, há uma listagem de rótulos dos vértices. Note que cada vértice possui um índice de 1 à  $n$ . Esse índice é importante, pois ele é utilizado nas definições das arestas. Depois da palavra “\*edges” cada linha conterá uma aresta. Por exemplo, na linha onde há “a b valor\_do\_peso”,  $a$  e  $b$  são os vértices que a aresta conecta, **valor\_do\_peso** é o peso da aresta.

Para o caso de grafos dirigidos, a palavra “\*arcs” aparece no lugar de “\*edges”.

```
*vertices n
1 rotulo_de_1
2 rotulo_de_2
...
n label_de_n
*edges
a b valor_do_peso
a c valor_do_peso
...
```