

Trabalho #01 - Grafos - representação em listas de adjacências

- **Data para entrega:** 09/05/2018 (até 23h55), via moodle.

Apresentação obrigatória para a profª, que definirá a ordem de apresentação.

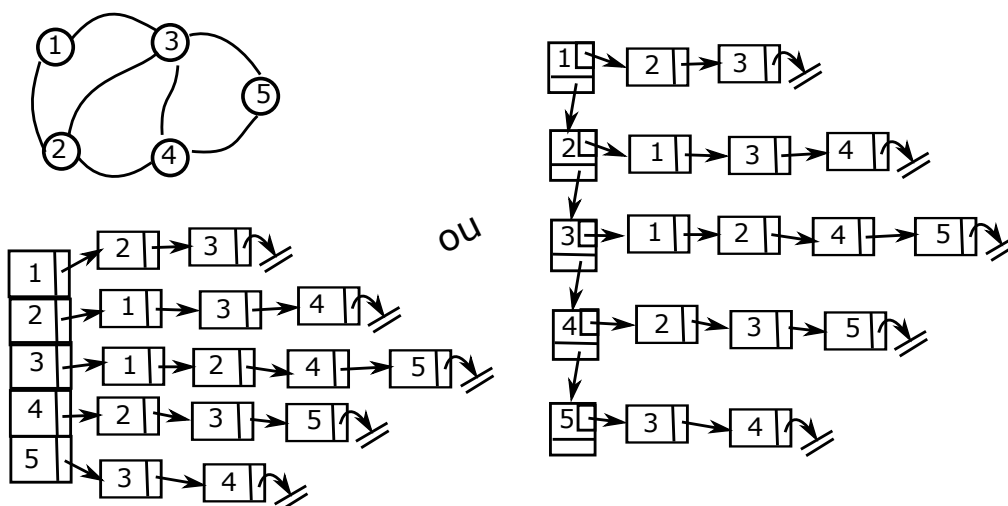
Serão descontados **2 pontos por dia de atraso**, com data limite até 13/05 às 23h55 (via moodle)!

Em caso de **cópia** de código, **os alunos envolvidos** terão nota igual a **zero** no trabalho 1 e em todas as avaliações restantes na disciplina, ou seja, T2, P2 e APS.

Enunciado do problema:

Faça um programa em **linguagem C** que represente internamente um dado grafo **simplex** e **não orientado**, $G = (V, E)$, como listas de adjacências.

Por exemplo, considere o grafo ilustrado na figura abaixo, você poderá optar por implementar um vetor estático de uma estrutura que contem o endereço do início de cada lista de adjacência, ou optar pela implementação de uma lista ligada (vertical) com cada nó contendo seus apontadores para suas respectivas listas de adjacência.



O programa deve conter as seguintes opções de menu:

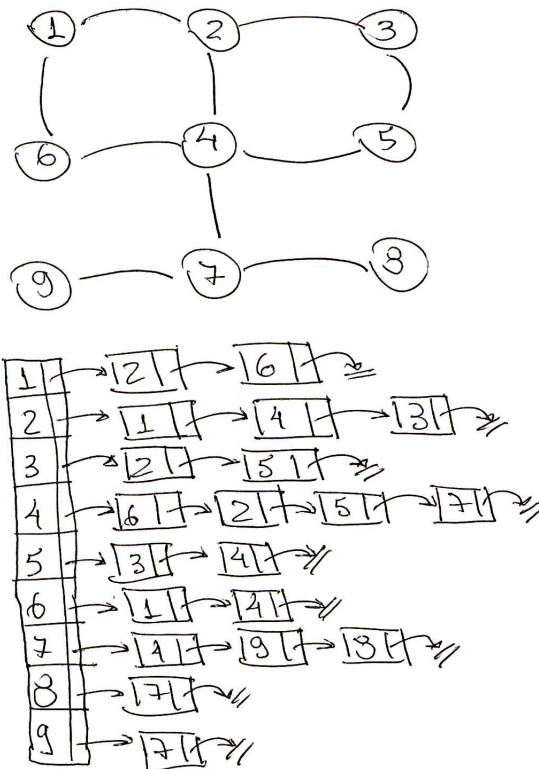
1. Inserir Vértices
2. Inserir Arestas
3. Visualizar Grafo.
4. Remover Vértices
5. Remover Arestas
6. Sair

Lista de itens que serão avaliados:

- Utilizou estrutura de dados em listas de encadeadas (alocação dinâmica de memória) para representar o grafo?
- Liberou a memória alocada?
- Inseriu e removeu vértices no grafo corretamente? Permitiu a inserção de vértices com identificação repetida?
- Inseriu e removeu arestas do grafo corretamente? Permitiu a inserção de arestas repetidas?
- Visualiza-se de forma clara e correta o grafo?
- A entrada de dados e saída de dados é amigável?
- Implementou funções parametrizadas?

Ex. `inserirVertice(int vert), inserirAresta(int vert1, int vert2).`

- O programa travou em tempo de execução?
- O programa está legível (identado) e bem documentado (comentado)?
- Permite inclusão do grafo (**via código**)? Por exemplo, considera o grafo abaixo:



Para inserí-lo via código, as seguintes chamadas de funções seriam declaradas:

```
inserirVertice(1);
inserirVertice(2); ...
inserirVertice(9);
inserirAresta(1,2);
inserirAresta(1,6); ...
inserirAresta(7,9);
```