

Trabalho Prático

1. Regras Básicas

1. Todos os programas devem ser desenvolvidos na linguagem C++.
2. Todas as apresentações de trabalho devem ser feitas no Linux usando o editor VIM.
3. Os trabalhos práticos têm como objetivo aperfeiçoar os conhecimentos do aluno sobre a matéria de Grafos e Teoria da Complexidade. Nos trabalhos, o aluno deve utilizar o Editor de Grafos disponibilizado pelo professor para implementar os conceitos mostrados em sala de aula.
4. O aluno deve comentar todos os métodos existentes no editor de grafos e os que forem acrescentados.
5. Para todos os problemas, o aluno deve fazer uma análise de complexidade de tempo para o pior caso. Essa análise não se resume apenas em jogar a ordem de complexidade. A análise deve ser colocada no próprio código fonte nos cabeçalhos dos métodos referentes a cada um dos problemas.
6. A estrutura de dados será a matriz de adjacências, mas ela será substituída durante o semestre. Somente os métodos *inserirAresta*, *isAresta*, *getAresta* e *excluirAresta* podem acessar a estrutura de dados.
7. Os programas serão entregues no SGA observando a entrada e a saída solicitadas.
8. Os trabalhos deverão ser apresentados ao professor na aula de laboratório, usando Linux.
9. Qualquer problema devido ao uso de outras arquiteturas será de responsabilidade do aluno.
10. Todas as classes devem ser entregues em um único arquivo, cujo nome será grafo.cc.
11. Atenção:
 - As cópias de trabalho serão encaminhadas para o colegiado.
 - O aluno perderá 1 ponto para cada método não comentado ou com comentário inútil.

12. Nos exercícios de ordenação, caso dois posts tenham a mesma chave de pesquisa, ordene-os pela ordem alfabética de seus textos.

2. Entrada Padrão:

A entrada contém vários casos de teste, sendo que cada um deles corresponde a um grafo. A primeira tarefa consiste em entender o exemplo contido no arquivo *entradaPadrao.in*. No caso dos exercícios sobre dígrafo, considere o arquivo *entradaPadraoDigrafo.in*.

3. Primeira Parte:

1. **Número de vértices e número de arestas:** Para cada grafo de entrada, imprima uma linha de saída contendo o número de vértices e o de arestas.
2. **Número de vértices pendentes e isolados:** Para cada grafo de entrada, imprima uma linha de saída contendo o número de vértices pendentes e o de isolados.
3. **Número de componentes:** Para cada grafo de entrada, imprima uma linha de saída contendo o número de componentes.
4. **Grau dos vértices e adjacentes:** Para cada vértice de cada grafo, imprima uma linha de saída contendo o grau do vértice e o índice de cada vizinho.
5. **Simples, regular, nulo, completo, euleriano e unicursal:** Para cada grafo de entrada, imprima uma linha de saída informando se o grafo é ou não simples, regular, nulo, completo, euleriano e unicursal.
6. **Grafo complementar:** Para cada grafo de entrada, criar um grafo complementar (cujo peso das arestas deve ser um) utilizando o mesmo padrão de entrada, contudo, sem o número de vértices.
7. **Grafo bipartite:** Para cada grafo de entrada, imprima uma linha de saída informando se o grafo é bipartite.

4. Segunda Parte:

Altere o editor de grafos para trabalhar com lista de adjacências e implemente as funcionalidades a seguir:

1. **Número de vértices e número de arestas:** Para cada grafo de entrada, imprima uma linha de saída contendo o número de vértices e o de arestas.
2. **Número de vértices pendentes e isolados:** Para cada grafo de entrada, imprima uma linha de saída contendo o número de vértices pendentes e o de isolados.
3. **Número de componentes:** Para cada grafo de entrada, imprima uma linha de saída contendo o número de componentes.
4. **Grau dos vértices e adjacentes:** Para cada vértice de cada grafo, imprima uma linha de saída contendo o grau do vértice e o índice de cada vizinho.
5. **Simplex, regular, nulo, completo, euleriano e unicursal:** Para cada grafo de entrada, imprima uma linha de saída informando se o grafo é ou não simples, regular, nulo, completo, euleriano e unicursal.
6. **Grafo complementar:** Para cada grafo de entrada, criar um grafo complementar (cujo peso das arestas deve ser um) utilizando o mesmo padrão de entrada, contudo, sem o número de vértices.
7. **Grafo bipartite:** Para cada grafo de entrada, imprima uma linha de saída informando se o grafo é bipartite.

5. Terceira Parte:

1. **Dígrafo:** Leia um arquivo (através de entrada padrão) contendo vários DÍGRAFOS e para cada um deles, gere uma linha de saída informando se o dígrafo é regular (*isRegular()*), balanceado (*isBalanceado()*), fortemente conexo (*isFortementeConexo()*) e euleriano (*isEuleriano()*). Nesse caso, para cada dígrafo, teremos seu número de vértices e um "quadrado" (matriz completa) com suas arestas.
2. **Busca em Profundidade:** Para cada grafo de entrada, imprima uma linha de saída contendo a ordem em que os vértices do grafo são visitados, sendo que o vértice 0 é o inicial e, em cada vértice, o de menor índice será sempre o primeiro a ser visitado.

3. **Busca em Largura:** Para cada grafo de entrada, imprima uma linha de saída contendo a ordem em que os vértices do grafo são visitados, sendo que o vértice 0 é o inicial e, em cada vértice, o de menor índice será sempre o primeiro a ser visitado.
4. **Dijkstra em Dígrafo:** Para cada grafo de entrada, imprima uma linha de saída contendo a distância do primeiro vértice em relação aos demais. Caso um vértice não esteja no mesmo componente do primeiro vértice considere que a distância entre eles é -1.

6. Quarta Parte:

1. **Árvore:** Para cada grafo de entrada, imprima uma linha de saída informando se o grafo é uma árvore ou não.
2. **AGM obtida pelo algoritmo de Prim:** Para cada grafo conexo de entrada, imprima uma linha de saída contendo a ordem que o algoritmo de Prim insere na AGM, onde o vértice inicial é o vértice 0. Se tivermos duas arestas com o mesmo peso, escolha aquela cuja soma dos índices dos vértices seja menor.
3. **AGM obtida pelo algoritmo de Kruskal:** Para cada grafo conexo de entrada, imprima uma linha de saída contendo a ordem que o algoritmo de Kruskal insere na AGM, onde o vértice inicial é o vértice 0. Se tivermos duas arestas com o mesmo peso, escolha aquela cuja soma dos índices dos vértices seja menor.
4. Para cada grafo conexo de entrada, imprima uma linha de saída informando o número de cut-vértices.