



ENUNCIADO

O trabalho consiste na implementação (em C/C++, JAVA ou C#) de alguns dos algoritmos vistos em sala de aula:

- | | | |
|-----------------------------|----------------------------------|---------------------------------|
| 1. Busca em Profundidade | 6. Algoritmo de Kruskal | 11. Algoritmo de Brélaz |
| 2. Busca em Largura | 7. Algoritmo de Fleury | 12. Algoritmo de Kosaraju |
| 3. Algoritmo de Dijkstra | 8. Algoritmo de König-Egerváry | 13. Algoritmo de Kahn |
| 4. Algoritmo de Jarník-Prim | 9. Algoritmo Guloso de Coloração | 14. Algoritmo de Bellman-Ford |
| 5. Ordenação Topológica | 10. Algoritmo de Welsh-Powell | 15. Algoritmo de Ford-Fulkerson |

Dessa forma, cada grupo possuirá **CINCO** integrantes onde cada um(a) deverá implementar (em comum acordo entre todos) exatamente três algoritmos da lista acima.

AVALIAÇÃO

O grupo terá um tempo total de **20 minutos** e deverá realizar o download do código **SUBMETIDO** no prazo devido no ambiente Canvas para avaliação. Todos os(as) integrantes **DEVERÃO** estar presentes durante a avaliação. Caso contrário, e sem a devida justificativa e comprovação, a nota será **ZERO** para os(as) alunos(as) faltantes. A avaliação envolverá o sorteio de **TRÊS** algoritmos da lista e serão realizados **DOIS** testes para validação de cada um dos métodos selecionados. Frisa-se que o(a) aluno(a) responsável pela implementação **NÃO DEVERÁ** auxiliar os(as) demais nas eventuais perguntas e questionamentos feitos pelo professor sobre o código e/ou sua execução. A rubrica da avaliação será: **50%** da nota para as execuções e resoluções corretas dos métodos e **50%** da nota para as explicações.

REQUERIMENTOS DO PROGRAMA

O programa deve ler (por meio do redirecionamento de entrada) um arquivo texto com as informações do usuário. Este arquivo contém um número de 1 a 15, correspondendo ao algoritmo desejado (conforme a tabela anterior) e o grafo descrito na seguinte forma: primeiro, têm-se a quantidade N de vértices do grafo e, depois, sua matriz de adjacência $N \times N$. Por exemplo, a entrada abaixo executará uma busca em profundidade no grafo bipartido completo $K_{3,3}$:

```
1
6
0 0 0 1 1 1
0 0 0 1 1 1
0 0 0 1 1 1
1 1 1 0 0 0
1 1 1 0 0 0
1 1 1 0 0 0
```

Para os algoritmos que necessitam da seleção do vértice inicial pelo usuário, pode-se selecionar o primeiro vértice do grafo dado como tal. Para todo e qualquer algoritmo selecionado, deve-se imprimir a matriz de adjacência do grafo dado. Além disso, a saída de cada método deverá ser clara e intuitiva para o entendimento do usuário de que o programa está executando apropriadamente. Por exemplo, para a entrada anterior, podemos escrever a ordem de visita dos vértices e resultando na saída indicada abaixo:

```
0 0 0 1 1 1
0 0 0 1 1 1
```



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Instituto de Ciências Exatas e Informática

Trabalho Prático

Algoritmos em Grafos (32511.00) – PBR (Noite) – 1º/2024

Prof. Felipe de Castro Belém (*felipebelem@pucminas.br*)

0 0 0 1 1 1

1 1 1 0 0 0

1 1 1 0 0 0

1 1 1 0 0 0

Ordem de Visita: 0,3,1,4,2,5