

QUARTO TESTE

Universidade Federal de Goiás (UFG) - Regional Jataí
Bacharelado em Ciência da Computação
Teoria de Grafos
Esdras Lins Bispo Jr.

22 de agosto de 2017

ORIENTAÇÕES PARA A RESOLUÇÃO

- A avaliação é individual, sem consulta;
- A pontuação máxima desta avaliação é 10,0 (dez) pontos, sendo uma das 06 (seis) componentes que formarão a média final da disciplina: quatro testes, uma prova e os exercícios de aquecimento;
- A média final (MF) será calculada assim como se segue

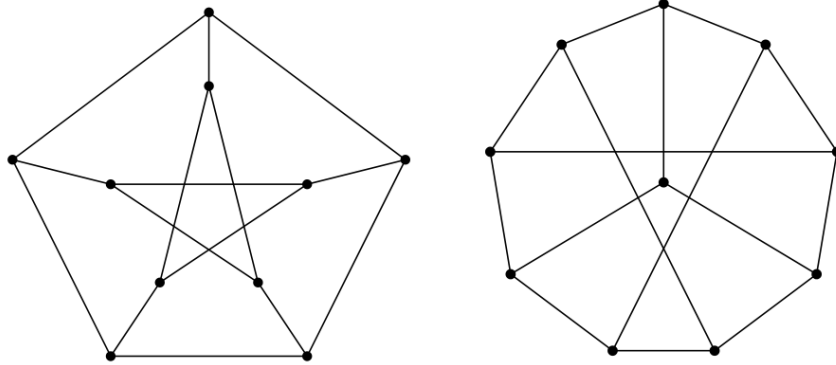
$$MF = MIN(10, S)$$
$$S = \left(\sum_{i=1}^4 0,2.T_i\right) + 0,2.P + 0,1.EA$$

em que

- S é o somatório da pontuação de todas as avaliações,
 - T_i é a pontuação obtida no teste i ,
 - P é a pontuação obtida na prova, e
 - EA é a pontuação total dos exercícios de aquecimento.
- O conteúdo exigido compreende os seguintes pontos apresentados no Plano de Ensino da disciplina: (5) Cortes e Pontes, (6) Árvores, (7) Isomorfismo e (9) Planaridade.

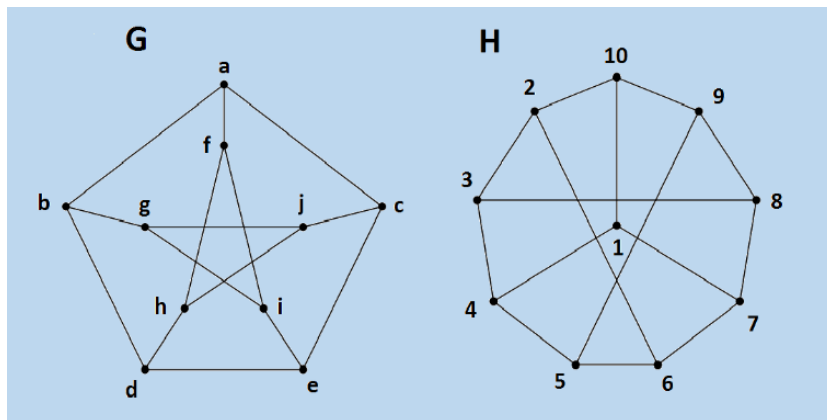
Nome:

1. (5,0 pt) [E 2.7] Os grafos abaixo são isomorfos? Justifique.



Resposta:

Sim, são isomorfos. Seja a rotulação dos vértices dos dois grafos conforme se segue:



É possível estabelecer uma bijeção entre V_G e V_H de forma a preservar as relações de adjacências entre os grafos. A bijeção é dada a seguir:

V_G	a	b	c	d	e	f	g	h	i	j
V_H	10	2	9	3	8	1	6	4	7	5

Como foi possível estabelecer a bijeção, os dois grafos são isomorfos ■

2. (5,0 pt) Na linguagem C, complete as lacunas da função `contemCircuito` conforme apresentada abaixo. Você deve substituir apenas as linhas 6 e 9, pelos trechos de código 1 e 2, respectivamente. O objetivo é que esta função retorne valor 1, se o grafo fornecido como parâmetro contiver um circuito. O valor de retorno será 0, caso não for.

```
1 int contemCircuito(Grafo *g){
2
3     for(int v=0; v<g->n; v++)
4         for(int w=0; w<g->n; w++)
5             if( g->adj[v][w] == 1){
6                 // TRECHO 1
7             }
8
9         //TRECHO 2
10
11 }
```

Resposta:

```
1 //TRECHO 1
2 if(ehPonte(g, v, w) == 0)
3     return 1;
```

```
1 //TRECHO 2
2 return 0;
```

1 grafos.h

```
1  #ifndef GRAFO_H_INCLUDED
2  #define GRAFO_H_INCLUDED
3
4  #include <stdlib.h>
5
6  typedef struct grafo {
7      char *nome;
8      int n;          // Numero de vertices
9      int m;          // Numero de arestas
10     int **adj;       // Ponteiro para a matriz de adjacencias
11 } Grafo;
12
13 int **gerarMatriz(int n);
14 void inicializar(Grafo *g, char *nome, int n);
15 void inserirAresta(Grafo *g, int v, int w);
16 void removerAresta(Grafo *g, int v, int w);
17 void exibir(Grafo *g);
18 int grau(Grafo *g, int v);
19 int grauMinimo(Grafo *g);
20 int grauMaximo(Grafo *g);
21 int ehRegular(Grafo *g);
22 int eh3Regular(Grafo *g);
23 Grafo *gMenosV(Grafo *g, int v);
24 int ehCaminho(Grafo *g);
25 int ehConexo(Grafo *g);
26 int ehPonte(Grafo *g, int v, int w);
27
28 #endif // GRAFO_H_INCLUDED
```

2 fabrica.h

```
1  #ifndef FABRICA_H_INCLUDED
2  #define FABRICA_H_INCLUDED
3
4  #include <stdlib.h>
5  #include "grafo.h"
6
7  Grafo *k(int n);
8  Grafo *caminho(int n);
9  Grafo *circuito(int n);
10
11 #endif // FABRICA_H_INCLUDED
```