

# **Lista de Exercícios - Algoritmos em Grafos**

**Universidade Federal do Ceará - Campus Sobral**

**Curso de Engenharia da Computação**

**Professor: Josefran Bastos**

(30) (TRABALHO OPTATIVO: Ajuda na AP1 ) Sabemos que não se conhece uma forma eficiente para verificar se dois grafos são isomorfos. Por outro lado, sabemos que grafos que possuem certas características diferentes (como número de arestas diferentes) são obviamente não isomorfos. Com isso, estamos interessados em gerar um pré-processamento para avaliar inicialmente essas características. Assim,

- (a) Pense em ao menos 3 características (para ganhar pontuação mínima) ou pelo menos 5 características (para ganhar pontuação máxima) que sejam “rápidas” para verificar.
  - (Obs 1:) Uma forma de saber se a característica que você está verificando é rápida de verificar é só gerar um grafo aleatório com algo em torno de 10 15 vértices e testar. Se demorar mais que 1h para terminar é porque não é tão rápida.
  - (Obs 2:) Caso você tenha domínio, você pode argumentar que seu algoritmo não é exponencial para argumentar que ele é rápido.
- (b) Para cada característica escreva uma função em C ou C++ que recebe dois grafos (representados como descritos no item (30)c) e retorna 1 se eles são diferentes nesta característica ou 0 caso contrário;
- (c) O seu programa não deverá acessar as arestas diretamente. Ao invés disso, deverá fazer uso das seguintes funções e estruturas

```
struct Grafo{
    int tamanho;
    int adjMatrix[tamanho*(tamanho-1)/2];
}
```

```
int edge(int i, int j, struct Grafo *g){
    return g->adj[getIndex(i, j, g->tamanho)];
}
```

```
void setEdge(int valor, int i, int j, struct Grafo *g){
    g->adj[getIndex(i, j, g->tamanho)] = valor;
}
```

•

- (d) Crie o algoritmo que verifica dois grafos são isomorfos usando as características desenvolvidas.

Por fim, usando o grafo  $G(n, p)$  (para  $p = \frac{1}{4}, \frac{2}{4}$  e  $\frac{3}{4}$ ) e a biblioteca time, calcule e compare o tempo médio de execução do seu algoritmo com o algoritmo da Questão (20). Compare então o seu algoritmo com ele mesmo apenas alternando as ordens das características.

**OBS** Calcular o tempo médio de execução do seu algoritmo significa realizar o procedimento da Questão (14) para uma quantidade fixa de repetições (assumiremos aqui que serão feitas 100 repetições).

```

int getIndex(int i, int j, int n){
    if( $i < j$ )
        return  $(2 * n - i - 1) * i / 2 + (j - i) - 1$ ;
    else
        return  $(2 * n - j - 1) * j / 2 + (i - j) - 1$ ;
}

```

Usando as funções que você desenvolveu, crie uma função que testa se dois grafos são isomorfos. Compare ele com a versão original. Alterne a ordem de verificação das características.

**O que entregar:** Deverá ser entregue um texto explicando cada característica, juntamente com a prova/argumento que o fato dela ser diferente implica no não isomorfismo. O texto deverá acompanhar o código e as tabelas de comparações geradas. A tabela deverá conter cada algoritmo com o tempo médio para  $n = 8, 12$  e  $16$  e para os  $p$  acima descritos.