# TRABALHO PRÁTICO 4:
# Fractal Roda

**Pedro Araujo Pires**

[1]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG)

ppires@dcc.ufmg.br

## 1. INTRODUÇÃO

Neste trabalho foi implementado um algoritmo que determina se um grafo é um fractal roda.

Um grafo roda é um grafo não direcionado onde existe um vértice central que é adjacente a todos os outros vértices do grafo, e o subgrafo desses outros vértices formam um ciclo. Um fractal roda consiste em um grafo roda central, onde cada vértice adjacente ao central também é um vértice central de um grafo roda. Esse fractal pode crescer indefinidamente, fazendo com cada vértice adjacente a um vértice central de um grafo roda também seja o vértice central de um novo grafo roda.

## 2. SOLUÇÃO PROPOSTA

Para resolver o problema, foram utilizadas algumas propriedades dos vértices de um fractal roda, que são:

- O vértice central do fractal possui grau $h$.
- Os vértices da borda do fractal possuem grau 3.
- Todos os demais vértices possuem grau $h + 3$.

Com essas propriedades, é fácil identificar qual é o vértice central do fractal (caso o grafo seja mesmo um fractal). Sabendo qual é o vértice central, o grafo é percorrido recursivamente, verificando se os vértices adjacentes do vértice central também são vértices centrais de grafos roda. Ao chegar na borda do grafo, a busca termina. Se em qualquer momento for encontrado um vértice que deveria fazer parte de uma roda, mas não faz, então o grafo não é um fractal roda.

Ao final da execução, o número de grafos roda visitados é salvo. De posse desse número, é possível saber se esse número é compatível com o número de grafos roda em um fractal de profundidade $d$. Foi determinado empiricamente que o total de grafos roda em um fractal roda é dado por

$$\sum_{k=0}^{d} h^k$$

onde $h$ é o grau do vértice central (tamanho dos grafos roda que compõem o fractal) e $d$ é a profundidade do fractal.

Existem porém algumas exceções, que ocorrem quando o fractal é composto por grafos roda de tamanho 3, ou quando ele tem profundidade 0. Quando um fractal é composto por grafos roda de tamanho 3, tanto o vértice central quanto os vértices da borda do fractal possuirão grau 3, impossibilitando a determinação do vértice central. Quando

isso ocorre, é verificado se o total de vértices na borda do grafo é uma potência de 3. Os vértices na borda de um fractal sempre são uma potência de $h$.

Quando um fractal possui profundidade 0, significa que ele é composto somente por 1 grafo roda, e existirão somente vértices com dois graus diferentes: $n$ e 3. Basta então verificar se o grafo é uma roda. Este caso também possui uma exceção, que é quando o grafo é uma roda de grau 3. Neste caso, é verificado se o grafo é um grafo completo, pois o grafo completo de 4 vértices é o grafo roda de grau 3.

## 2.1. Estruturas de dados

### 2.1.1. Grafo:

Armazena os dados de um grafo.

| **1**: Grafo |
|---|
| matriz de adjacência; |
| total de vértices; |
| total de arestas; |
| *array* com os graus de todos os vértices; |

As estruturas de dados foram alocadas dinamicamente. O programa `valgrind` foi utilizado para verificar se toda a memória utilizada pelo programa foi desalocada corretamente. A saída do programa foi:

```
HEAP SUMMARY:
    in use at exit: 0 bytes in 0 blocks
  total heap usage: 2,816 allocs, 2,816 frees, 4,732,587 bytes allocated

All heap blocks were freed -- no leaks are possible
```

## 2.2. Análise de Complexidade

Inicialmente é calculado quantos graus diferentes os vértices do grafo possuem. A função que faz esse cálculo percorre o *array* com os graus dos vértices, e vai armazenando em outro *array* os graus já encontrados. Para cada vértice, o *array* com os graus já encontrados é percorrido. Sendo $g$ a quantidade de graus diferentes no grafo, e $V$ o número de vértices, essa função possui complexidade de tempo $O(gV)$. Também é calculado quantos vértices de grau 3 o grafo possui. A função que faz esse cácluclo somente percorre o *array* de graus. Logo, a complexidade de tempo é $O(V)$.

Em seguida é verificado quantos graus diferentes os vértices do grafo possuem. Caso esse número seja 1 ou 2, significa que o grafo pode ser um fractal de grau 3, e as verificações necessárias são feitas. Se for 3 (caso mais comum), primeiro é encontrado o vértice com o grau intermediário. Para isso, o *array* com os graus é percorrido 3 vezes, e a função que faz esse cálculo possui complexidade $O(3V) = O(V)$.

Em seguida é chamada uma função recursiva que verifica se um vértice é um vértice central de um grafo roda. Caso ele seja, a função é chamada novamente para cada vértice adjacente ao vértice central. A cada vez que a função é executada, primeiro é feita a verificação do grafo roda. Essa verificação primeiro percorre todos os adjacentes

ao vértice central, e para cada um, percorre todos os outros adjacentes (no pior caso). Logo, ela é $O(d^2)$, onde $d$ é o grau do grafo roda. Depois, todos as arestas do grafo roda verificado são retirados do grafo. Essa função é similar à função de verificação do grafo roda, e possui a mesma complexidade $O(d^2)$. Logo, a complexidade dessa função é $O(2d^2) = O(d^2)$, para cada vez que ela é executada. Como ela é executada uma vez para cada grafo roda presente no fractal, e um fractal roda possui $(h^d - 1)/(h - 1)$ grafos roda (soma de uma progressão geométrica), a verificação de um fractal roda possui complexidade $O(d^2 h^d)$

# 3. IMPLEMENTAÇÃO

## 3.1. Código

### 3.1.1. Arquivos .c

- **io.c:** Define as funções relacionadas à parte de entrada e saída do programa.
- **benchmark.c:** Define as funções relacionadas à medição do tempo de execução.
- **grafo.c:** Define as funções relacionadas à strutura de dados Grafo.
- **fractal.c:** Define as funções relacionadas ao algoritmo para verificar um fractal.
- **main.c:** Define a função main do programa.

### 3.1.2. Arquivos .h

- **io.h:** Define os cabeçalhos das funções relacionadas à parte de entrada e saída do programa.
- **benchmark.h:** Define os cabeçalhos das funções relacionadas à medição do tempo de execução.
- **boolean.h:** Define o tipo de dado boolean.
- **grafo.h:** Define os cabeçalhos das funções sobre o tipo de dado Grafo, e a struct Grafo.
- **fractal.h:** Define os cabeçalhos das funções contidas no arquivo fractal.c.

## 3.2. Compilação

O programa deve ser compilado através do comando `make`. Esse comando irá compilar todos os arquivos, e gerar um executável chamado `tp4`.

## 3.3. Execução

A execução dos programas recebe como parâmetro somente o nome do arquivo que contém os dados de entrada. O comando para a execução do programa é da forma:

```
./tp4 <arquivo de entrada> [benchmark]
```

O parâmetro opcional $benchmark$, se passado, fará com que ao final da execução de cada instância do problema, seja impresso na tela o tempo de execução (em segundos) daquela instância.

### 3.3.1. Formato da entrada

O arquivo com os dados de entrada possui várias instâncias do problema. Cada instância ocupa duas linhas do arquivo, onde a primeira contém um número inteiro, que indica quantos vértices o grafo possui, e a segunda linha contém todas as arestas do grafo. Cada aresta é representada por dois vértices separados por espaço, e cada aresta é separada por vírgula.

```
7
0 1,0 2,0 3,0 4,0 5,0 6,1 2,2 3,3 4,4 5,5 6,6 1
1000
0 999
```

### 3.3.2. Formato da saída

A saída do programa consiste em somente imprimir na tela a profundidade e o grau do fractal. Caso o grafo não seja um fractal roda, é impresso $-1$.

```
0 6
-1
```

## 4. AVALIAÇÃO EXPERIMENTAL

Para fazer a avaliação experimental do programa, foi utilizado o arquivo `tp4Tests_input.txt`, disponibilizado no *moodle* pelos monitores da disciplina. O programa foi executado 10 vezes, e a Tabela 1 mostra a média dos resultados dessas execuções.

| # | Tempo de execução($10^{-6} seg.$) |
|---|---|
| 1 | 644 |
| 2 | 44 |
| 3 | 885 |
| 4 | 5 |
| 5 | 6 |
| 6 | 91 |
| 7 | 306 |
| 8 | 8 |
| 9 | 11 |
| 10 | 23 |
| 11 | 5 |

**Tabela 1. Tempo de execução do arquivo** `tp4Tests_input.txt`

Todos os testes foram feitos em uma máquina com CPU Intel Core 2 T7400, com 2 processadores de 2.17GHz cada, rodando o sitema operacional Ubuntu Linux 10.04.

## 5. CONCLUSÃO

Neste trabalho foi implementado um algoritmo para verificar se um grafo é um fractal roda. O trabalho atingiu o objetivo proposto, que era a familiarização com os algoritmos em grafos.