



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Bacharelado em Ciência da Computação

Gabriel de Cortez Mourão
Luís Augusto Lima de Oliveira
Mateus Fernandes Barbosa
Victor Ferraz de Moraes

Teoria dos Grafos e Computabilidade: Implementação 2

Belo Horizonte

2024

Gabriel de Cortez Mourão
Luís Augusto Lima de Oliveira
Mateus Fernandes Barbosa
Victor Ferraz de Moraes

Teoria dos Grafos e Computabilidade: Implementação 2

Documentação para exercício da disciplina
de Teoria dos Grafos e Computabilidade do
curso de Ciência da Computação da Pontifi-
cia Universidade Católica de Minas Gerais.

Belo Horizonte

2024

SUMÁRIO

1	INTRODUÇÃO	1
2	DESCRIÇÃO DAS FUNÇÕES	2
2.1	Função fatorial	2
2.2	Função calcula_subgrafos	2
2.3	Função exhibe_grafo	3
2.4	Função gera_subgrafos	3
2.5	Função <i>main</i>	4

1 INTRODUÇÃO

Este documento descreve as funções implementadas em um código C++ que calcula subgrafos, exibe grafos, e gera subgrafos únicos com base em um número de vértices N . O documento abrange a funcionalidade de cada função, os tipos de variáveis utilizados, e a ordem de execução do código.

2 DESCRIÇÃO DAS FUNÇÕES

2.1 Função fatorial

- **Descrição:** Função que executa o fatorial de um determinado valor. É utilizada para efetuar a verificação matemática do total de subgrafos possíveis a partir de um conjunto de vértices.
- **Assinatura:** `int fatorial(int n);`
- **Parâmetros:**
 - `int n`: O número do qual se deseja calcular o fatorial.
- **Retorno:**
 - `int`: O fatorial de n .

2.2 Função calcula_subgrafos

- **Descrição:** Esta função calcula o número teórico de subgrafos possíveis para um grafo com N vértices. A fórmula utilizada combina a quantidade de combinações de vértices com a quantidade de possíveis arestas entre eles. Para cada quantidade p de vértices (de 1 até N), a função calcula:

$$\text{combinacoes} = \frac{N!}{p!(N-p)!}$$

e o número de arestas possíveis:

$$\text{arestas_possiveis} = 2^{\frac{p(p-1)}{2}}$$

O total de subgrafos é acumulado e retornado ao final.

- **Assinatura:** `int calcula_subgrafos(int N);`

- **Parâmetros:**

- *int* N: O número de vértices no grafo.

- **Retorno:**

- *int*: O número total de subgrafos possíveis.

2.3 Função `exibe_grafo`

- **Descrição:** Esta função exibe um grafo a partir de dois vetores booleanos: `vertices` e `arestas`. O vetor `vertices` indica quais vértices estão presentes no subgrafo, enquanto `arestas` indica as conexões entre os vértices. A função imprime os vértices ativos e as arestas presentes, além de mostrar os vetores de vértices e arestas de forma binária.

- **Assinatura:** `void exibe_grafo(const vector<bool> & vertices
const vector<bool> & arestas, int N)`

- **Parâmetros:**

- `const vector<bool> & vertices`: Um vetor booleano indicando quais vértices estão presentes.
- `const vector<bool> & arestas`: Um vetor booleano indicando quais arestas estão presentes.
- *int* N: O número de vértices no grafo.

- **Retorno:**

- `void`: Não há retorno, apenas exibe o grafo no console.

2.4 Função `gera_subgrafos`

- **Descrição:** Esta função gera todos os subgrafos possíveis de um grafo com *N* vértices e filtra aqueles que são válidos. Em seguida, exibe os subgrafos únicos utilizando a função `exibe_grafo`. Ela também compara a quantidade de subgrafos gerados com o valor teórico calculado pela função `calcula_subgrafos(int N)`.

- **Assinatura:** `void gera_subgrafos(int N);`

- **Parâmetros:**

- *int* N: O número de vértices no grafo.

- **Retorno:**

- *void*: Não há retorno, apenas exibe os subgrafos únicos no console.

2.5 Função *main*

- **Descrição:** A função *main()* é a porta de entrada do programa. Ela solicita ao usuário o número de vértices *N* e chama a função *gera_subgrafos(int N)* para iniciar o processo de geração e exibição dos subgrafos.

- **Assinatura:** *int main()*;

- **Parâmetros:**

- *void*: A função não recebe parâmetros.

- **Retorno:**

- *int*: Retorna 0 para indicar a execução bem-sucedida do programa.