Algoritmos e Programação II

https://evandro-crr.github.io/alg2

Recursão

Função que chama a si mesmo.

```
void mensagem() {
    cout << "Chamada Recursiva\n";
    mensagem();
}
```

```
void mensagem(int contador) {
    if (contador > 0) {
        cout << "Chamada Recursiva\n";
        mensagem(contador - 1);
    }
}
```

Por que usar recursão?

- **Pros**: Facilita a resolução de certos problemas.
- **Contras**: Não é necessário usar recursão e implementações recursivas podem ser menos eficientes

O **overhead** da implantação recursiva é *compensado*pela facilidade da implementação.

Divisão do problema

Na recursão, quebramos o problema em partes menores:

- **Caso base**: Se o problema for pequeno o suficiente, a função retorna a solução.
- **Caso recursivo**: Caso contrário, a função reduz o problema e chama a si mesma para resolver.

Função Fatorial

Definição

$$n! = \begin{cases} 1, & \text{se } n = 0 \\ 1 \times 2 \times 3 \times \cdots \times n, & \text{caso contrário} \end{cases}$$

Definição recursiva

$$n! = \begin{cases} 1, & \text{se } n = 0 \\ n \times (n-1)!, & \text{caso contrário} \end{cases}$$

Exemplo: Contagem de Caracteres

Implementação Iterativa

```
int contar(char letra,
           const char *inicio,
           const char *fim)
{
    int contador = 0;
    for (; inicio != fim; inicio++) {
        if (*inicio == letra) {
            contador++;
        }
    }
    return contador;
}
```

Implementação Recursiva

```
int contar_r(char letra,
             const char *inicio,
             const char *fim)
{
    if (inicio == fim) {
        return 0;
    } else if (*inicio == letra) {
        return 1 + contar_r(letra, ++inicio, fim);
    } else {
        return contar_r(letra, ++inicio, fim);
    }
}
```

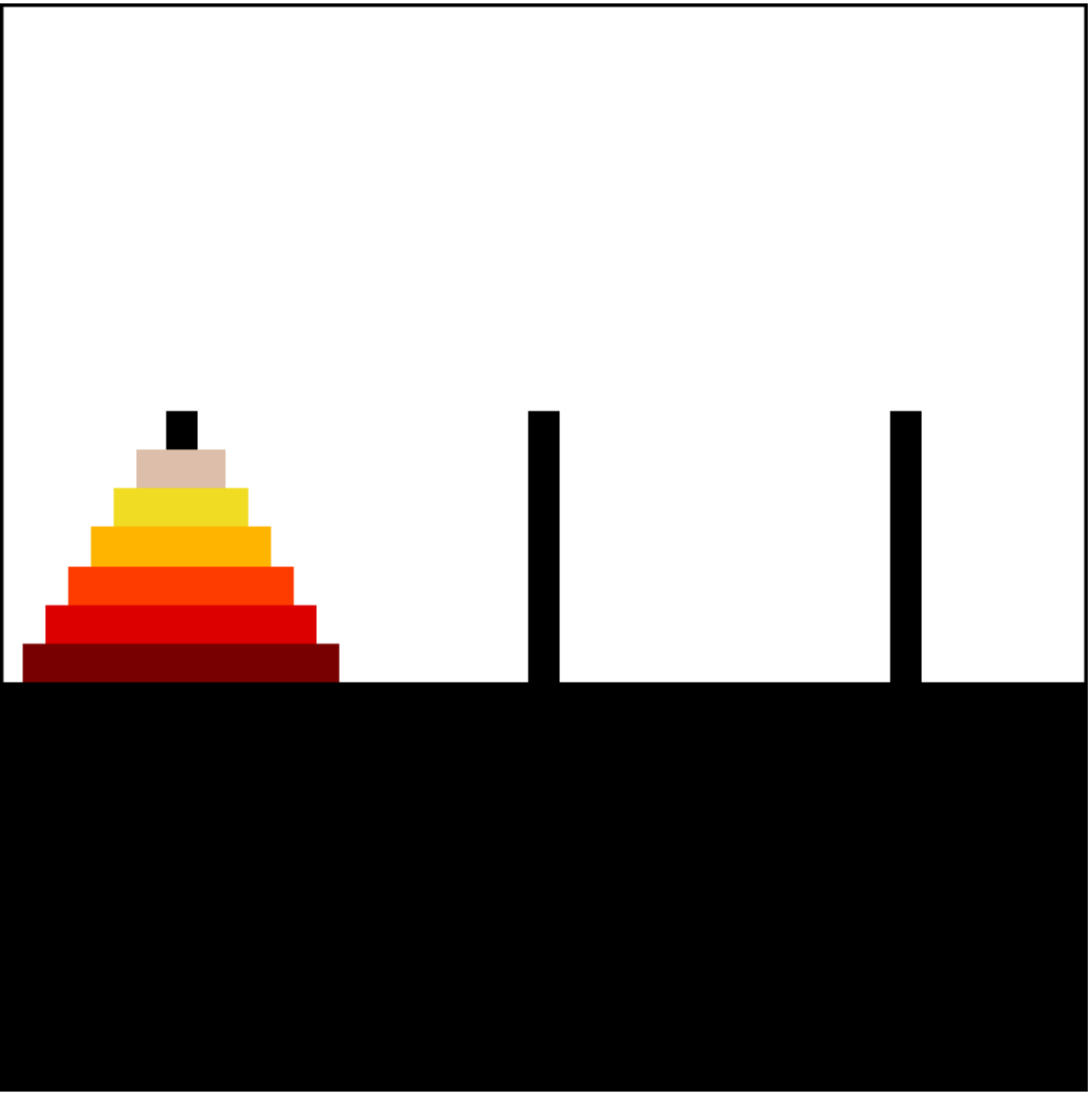
Máximo Divisor Comum (GCD)

Algoritmo de Euclides

$$\gcd(x, y) = \begin{cases} y, & \text{se } y \mid x \\ \gcd(y, x \mod y), & \text{caso contrário} \end{cases}$$

Sequência de Fibonacci

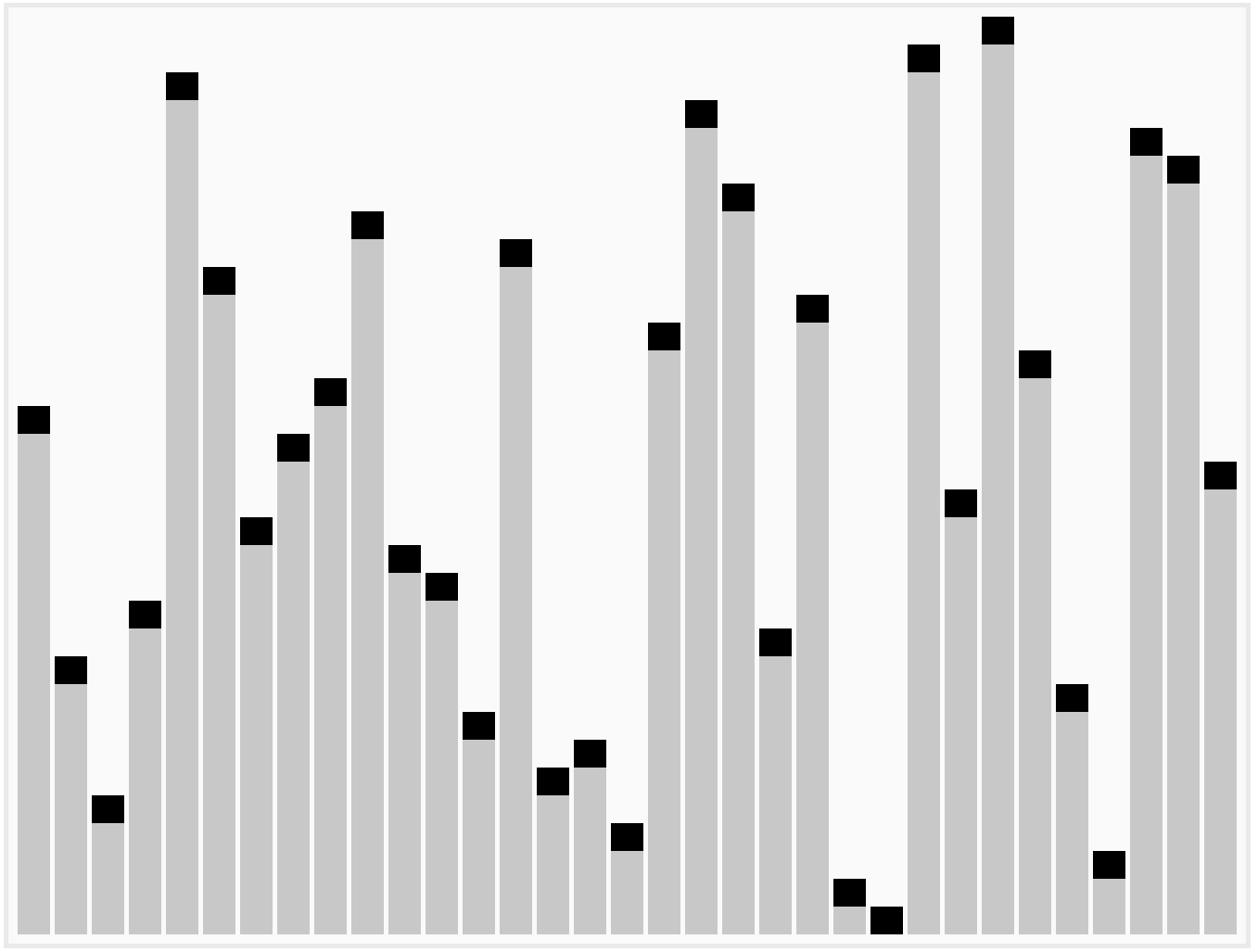
$$F(n) = \begin{cases} 0, & \text{se } n = 0 \\ 1, & \text{se } n = 1 \\ F(n-1) + F(n-2), & \text{se } n > 1 \end{cases}$$



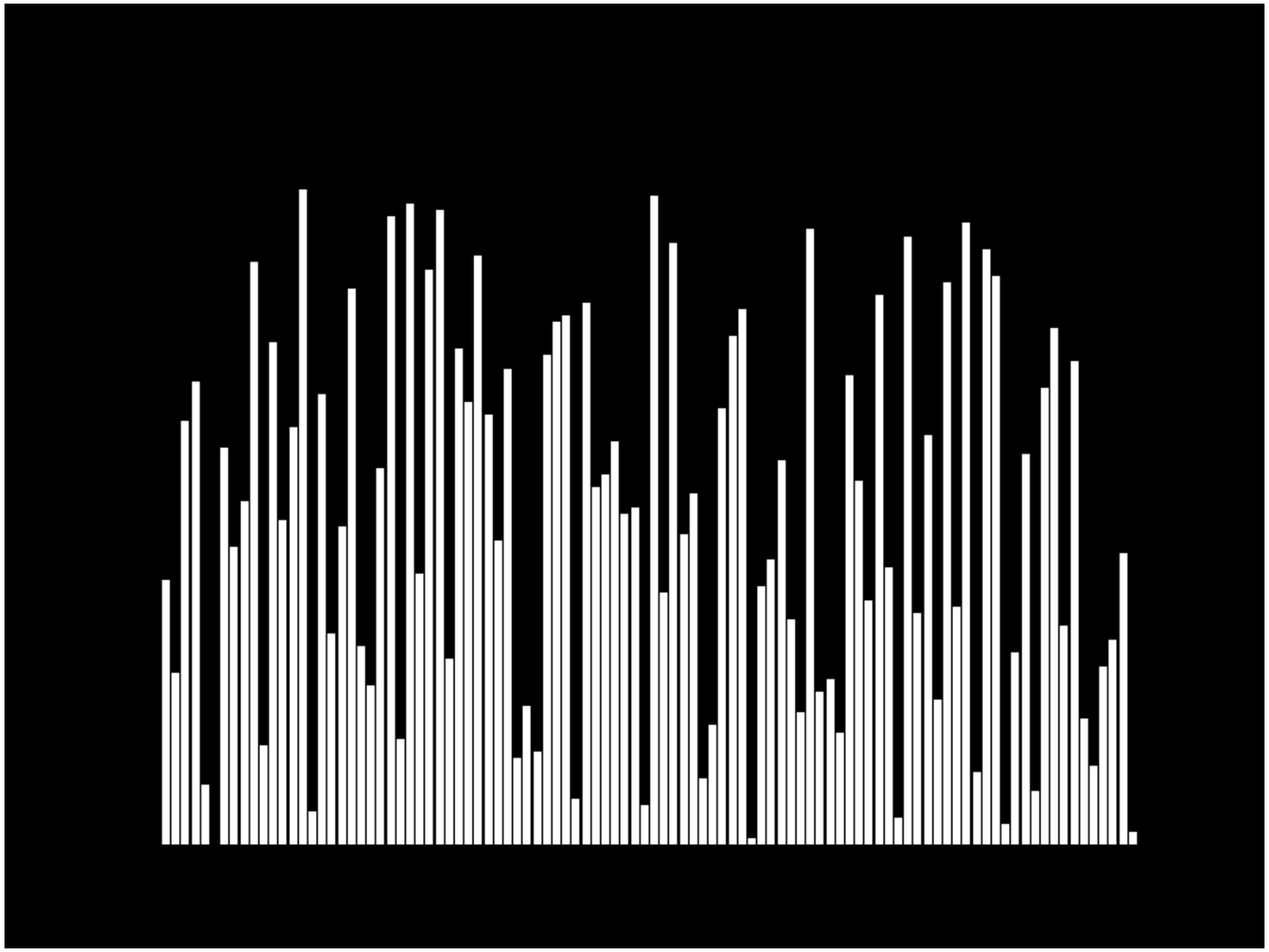
Torre de Hanoi

https://www.mathsisfun.com/games/towerofhanoi.html

Quick Sort



Merge Sort



Algoritmos e Programação II

https://evandro-crr.github.io/alg2

Evandro Chagas Ribeiro da Rosa