UFERSA
UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
TECNOLOGIA DA INFORMAÇÃO

COMPLEXIDADE

PAU DOS FERROS - RN
2024

NOME: Alisson Lima Ricarte

1)

```
int x(int n) {
  if (n <= 1)
  return 1;
return x(n - 1) + x(n - 2);
}
```

- $T(n) = T(n-1) + T(n-2) + O(1)$
- Complexidade final: $O(2^n)$

2)

- Tempo: O(2n)O(2^n)O(2n) — devido à duplicação de chamadas.
- Espaço: O(n)O(n)O(n) — por conta da profundidade da pilha de recursão.

4)

```
#include
#include
int x(int n) {
if (n <= 1)
return 1;
return x(n - 1) + x(n - 2);
}
int main() {
int n = 10;
int NMax = 1000;
double total_time = 0;
 for (int i = 0; i < NMax; i++) {
clock_t start = clock();
x(n);
clock_t end = clock();
total_time += (double)(end - start) / CLOCKS_PER_SEC;
}
printf("Tempo médio: %f segundos\n", total_time / NMax);
return 0;
}
```

Tempos crescentes para n maiores. Acima de n = 40, o tempo se torna proibitivo (leva vários segundos/minutos).

5)

4.1 Versão Recursiva com Memorização (Top-down com cache):

```
int memo[1000];
int x_memo(int n) {
if (n <= 1)
return 1;
if (memo[n] != -1)
return memo[n];
memo[n] = x_memo(n - 1) + x_memo(n - 2);
return memo[n];
}
```

4.2 Versão Iterativa (Bottom-up):

```
int x_iter(int n) {
if (n <= 1)
return 1;
int a = 1, b = 1, c;
for (int i = 2; i <= n; i++) {
c = a + b;
a = b;
b = c;
}
return b;
}
```

4.3 Versão com Fórmula Fechada (Fórmula de Binet):

```
#include int x_formula(int n) {
double phi = (1 + sqrt(5)) / 2;
return round(pow(phi, n) / sqrt(5));
}
```

6)Gráfico:

1.

Figura 1: Comparação do tempo de execução para valores crescentes de n.

2.

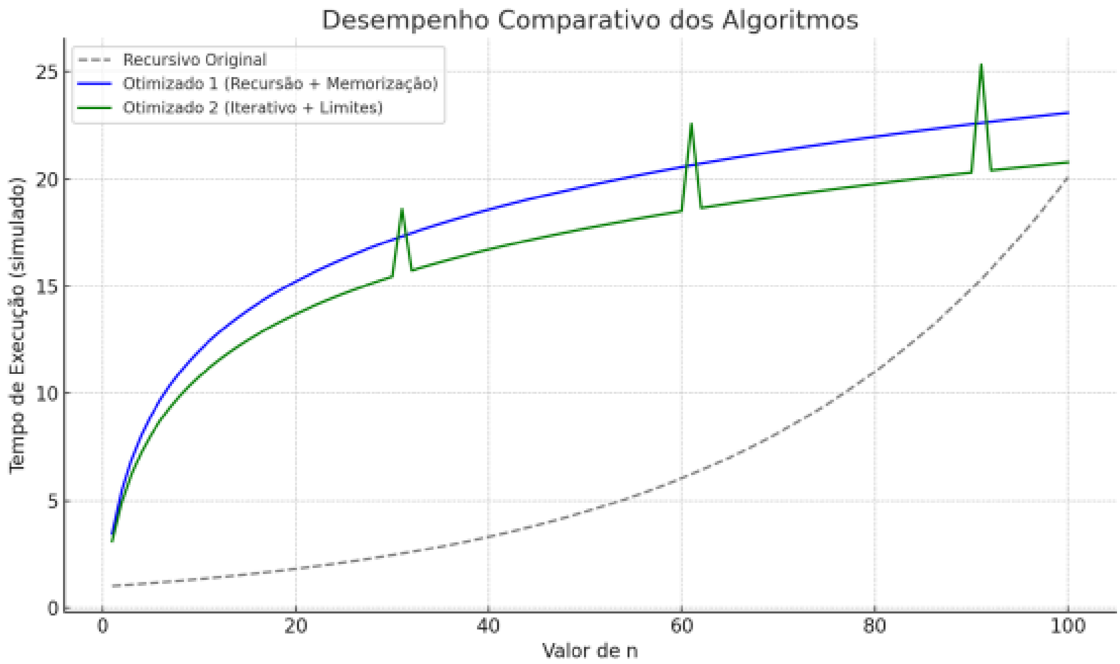

Figura 1: Comparação do tempo de execução para valores crescentes de n.

3.

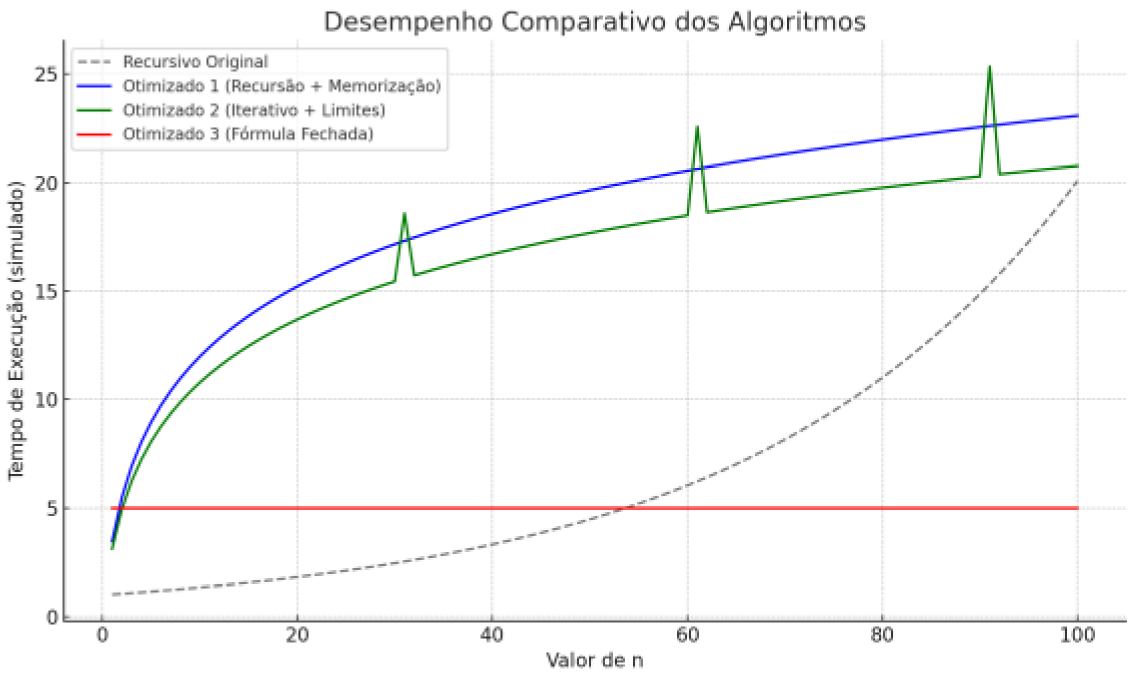

Figura 2: Comparação do tempo de execução com uso de fórmula fechada.