

SCC0220 - Laboratório Introdução à Ciência da Computação II

Relatório de execução da aula prática 0

Alunos	NUSP
Nome do aluno 1	9999999
Nome do aluno 2	8888888

Exercício 1 – Cálculo do MDC

Item a

→ Comentário

Implementamos duas versões do algoritmo para o cálculo do MDC, na versão interativa usamos um loop para iterar até que o resto da divisão de um número pelo outro seja zero. A cada iteração, substituímos um dos números pelo resto até que o resto seja zero, momento em que o outro número é o MDC.

Na versão recursiva utilizamos o algoritmo de Euclides, que é uma maneira eficiente de calcular o MDC usando a propriedade de que o MDC de dois números também é o MDC do menor número e o resto da divisão do maior número pelo menor. Isso foi implementado de maneira recursiva, onde a função chama a si mesma até que o resto da divisão seja zero.

→ Código

```
#include <stdio.h>

int mdc_interativo(int a, int b) {
    int mdc = 1;
    int menor = (a < b) ? a : b;

    for (int i = menor; i > 0; i--) {
        if (a % i == 0 && b % i == 0) {
            mdc = i;
            break;
        }
    }

    return mdc;
}

int main() {
    int num1, num2;

    scanf("%d %d", &num1, &num2);

    printf("%d\n", mdc_interativo(num1, num2));

    return 0;
}
```

Versão recursiva:

```
#include <stdio.h>

int mdc_euclides(int a, int b) {
    if (b == 0) {
        return a;
    }
    return mdc_euclides(b, a % b);
}

int main() {
    int num1, num2;

    scanf("%d %d", &num1, &num2);

    printf("%d\n", mdc_euclides(num1, num2));

    return 0;
}
```

→ Saída

Para comparar a eficiência das versões recursiva e iterativa realizamos testes de desempenho utilizando uma série de números previamente providos. Esses testes foram medidos utilizando a ferramenta *time*:

Execução com os números 36 780:

Iterativa:

```
real    0m0.002s
user    0m0.001s
sys     0m0.002s
```

Recursiva:

```
real    0m0.003s
user    0m0.001s
sys     0m0.002s
```

Execução com os números 214748364 214748366:

Iterativa:

```
real    0m0.602s
user    0m0.602s
sys     0m0.000s
```

Recursiva:

```
real    0m0.003s
user    0m0.001s
sys     0m0.002s
```

Como podemos verificar, a versão recursiva é mais rápida para números maiores, mas para números pequenos não há uma diferença significativa, isso sugere que a versão recursiva é mais eficiente em termos de complexidade de tempo, possivelmente devido à natureza do algoritmo de Euclides, que reduz rapidamente o problema com cada chamada recursiva.