CENTRO UNIVERSITÁRIO FEI
MESTRADO ENGENHARIA ELÉTRICA
ALGORITMOS COMPUTACIONAIS – PEL201

RELATÓRIO 1
MÁXIMO DIVISOR COMUM

GUSTAVO APARECIDO DE SOUZA VIANA – 120110-2
SÃO BERNARDO DO CAMPO
2020

O código fonte está disponível em:

https://github.com/apparecidoo/master-computational-algorithms

1)

Formalmente o MDC, Máximo Divisor comum, é o maior divisor que dois números naturais possam ter incluindo o número 1.

Para resolver este problema, foi utilizado a técnica de Euclides. Ele parte do princípio que o MDC não muda se o menor número for subtraído ao maior. Ou seja, o processo consiste em reduzir o maior número até que convergir em zero, quando isto acontecer o outro número maior de zero é o valor do MDC esperado.

2)

A solução iterativa é a execução de um conjunto operações inúmeras vezes até ter um momento de parar, podendo achar ou não a solução. No âmbito de programação utilizamos laços de repetição para fazer isto.

Abaixo a solução iterativa para resolver o problema de MDC por meio da técnica de Euclides.

```
function SolveIterative(int major, int minor) {
    double result = 1;
    while (result != 0) {
            result = major % minor;
            major = minor;
            minor = result;
    }
    return major;
}
```

3)

A solução recursiva possui a mesma ideia de repetição de operações, utilizando a chamada da mesma função com parâmetros atualizados ou diferentes e assim entrando em loop até que o “caso base” faça com que o loop pare, assim é possível achar ou não a solução.

Abaixo a solução recursiva para resolver o problema de MDC por meio da técnica de Euclides.

```
function SolveRecursive(int major, int minor) {
        if (minor == 0)
                return major;
        return SolveRecursive(minor, std::fmod(major,minor));
}
```

4)

A tabela abaixo mostra 20 tentativas feitas no algoritmo de Euclides, onde a coluna *first* e *second* são dois números gerados aleatoriamente. A coluna GCD é o resultado do MDC e por fim, *Time Iterative* e *Time Recursive* o tempo gasto em microssegundos para o processamento. Em seguida, é apresentado o gráfico comparando o tempo de execução da solução iterativa e recursiva, assim podemos ver que os tempos são bem parecidos, apresentando pequenas diferenças em alguns casos mas não pode concluir nada pois esses casos podem ser *outliers*.

Attempt	First	Second	GCD	Time Iterative	Time Recursive
1	82254502	49846935	1	7	2
2	87556380	70387466	2	1	2
3	10313593	713746	1	2	2
4	90595221	63739629	3	1	2
5	41354599	72893774	1	1	2
6	65356907	96083278	1	1	1
7	65617474	14428010	2	2	2
8	18526771	79141328	1	1	2
9	52353684	83467929	3	2	3
10	35504430	63979259	1	2	3
11	83355818	12786039	7	2	2
12	89778593	51063484	1	4	3
13	8368234	71438278	2	1	2
14	91453414	15492897	1	2	13
15	26830849	41147218	1	2	2
16	4728366	26544574	2	1	2
17	82422747	90016261	1	2	2
18	71786761	39567567	1	4	3
19	11025178	88992773	1	2	4
20	90497877	43387289	1	2	33