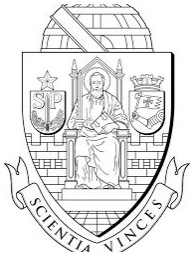

Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Introdução à Ciência de Computação II

Trabalho I

Donizeti Júnior **Nº USP: 9393882**

Docente: Ricardo J. G. B. Campello

São Carlos, Setembro de 2016

Introdução

O trabalho consistiu no desenvolvimento de um programa em C que, após receber um número binário, retornasse sua representação decimal. Foi necessária a construção de dois algoritmos distintos: um iterativo e um recursivo.

A seguir, temos os pseudocódigos de ambos os algoritmos usados na execução do problema:

Função iterativa:

```
Int bin2dec(int n) {
        declaração de variáveis
        int decimal <- 0                          1
        int i <- 0                                1

        Enquanto n != 0                           log(10)n
                aux <- (n % 10)                   2* log(10)n
                decimal += aux * 2^i              4* log(10)n
                n <- n / 10                       2* log(10)n
                i <- i + 1                        2* log(10)n

        retorne decimal                           1
}
```

T(n) = 1 + 1 + logn + 10*(logn) + 1 = 11*log(n) + 3
Portanto a complexidade é de **O(logn).**
Provando o resultado:

f(n) $\leq$ c*g(n), $\forall n \geq no$
11*log(n) + 3 $\leq$ c*log(n) e 11*log(n) + 3 $\leq$ log(n) + 3*log(n)
11*log(n) + 3 $\leq$ 4*log(n)

Assim, pegue c = 4 e n ≥ 2 e a complexidade Big-O está justificada.

Conclui-se, então, que a função log(10)n é limita superiormente a função que representa o pior caso do algoritmo utilizado a partir de c = 4.

Função recursiva:

```
Int bin2dec (int n) {
        se (n/10) = 0
                retorne  n
        senão
                retorne (n <- n % 10 + 2*bin2dec(n / 10))
}
```

$T(1) = T(0) = c$
$T(n) = 2*T(n/10) + c$, para $n \geq 10$
Portanto a complexidade é de **O(nlogn).**

Funcionamento

Ambos os algoritmos se concentram na ideia de somar, ao resultado final, o módulo do número binário digitado e então dividir o mesmo por 10 até que esta divisão valha 0.