Universidade Federal do Ceará – UFC
Centro de Ciências – CC
Mestrado e Doutorado em Ciências da Computação - MDCC
Estruturas de Dados Avançadas

Exercício: Complexidade de Algoritmos
Objetivos: Exercitar os conceitos de Complexidade de Algoritmos.
Data da Entrega: 19/03/2018

OBS 1: Exercício Indivudual.
OBS 2: A entrega desta lista deverá ser executada via SIGAA.

NOME: ISRAEL DE CASTRO VIDAL **MATRÍCULA:** 409500

Questão 1

As funções *f(n)* mostradas abaixo fornecem o tempo de processamento *T(n)* de um algoritmo resolvendo um problema de tamanho *n*. Complete a tabela abaixo colocando, para cada algoritmo, sua complexidade (*O maiúsculo*) e a ordem do mais eficiente para o menos eficiente. Em caso de empate repita a ordem (por exemplo: 1º ,2º, 2º, ...).

f(n)	$O(\dots)$	*ordem*
$5 + 0.001n^3 + 0.025n$	$O(n^3)$	9
$500n + 100n^{1.5} + 50n\log_{10} n$	$O(n^{1.5})$	5
$0.3n + 5n^{1.5} + 2.5 \cdot n^{1.75}$	$O(n^{1.75})$	6
$n^2 \log_2 n + n(\log_2 n)^2$	$O(n^2 \log n)$	8
$n\log_3 n + n\log_2 n$	$O(n \log n)$	2
$3\log_8 n + \log_2 \log_2 \log_2 n$	$O(\log n)$	1
$100n + 0.01n^2$	$O(n^2)$	7
$0.01n + 100n^2$	$O(n^2)$	7
$2n + n^{0.5} + 0.5n^{1.25}$	$O(n^{1.25})$	4
$0.01n\log_2 n + n(\log_2 n)^2$	$O(n.(\log n)^2)$	3
$100n\log_3 n + n^3 + 100n$	$O(n^3)$	9
$0.003\log_4 n + \log_2 \log_2 n$	$O(\log n)$	1

Questão 2

Os algoritmos abaixo são usados para resolver problemas de tamanho n. Descreva e informe para cada algoritmo sua complexidade no pior caso (O maiúsculo/Ômicron). Tente entender o problema antes de apresentar uma resposta.

a)

```
for ( i=1; i < n; i *= 2 ) {
    for ( j = n; j > 0; j /= 2 ) {
        for ( k = j; k < n; k += 2 ) {
            sum += (-j * k) << i/2;
        }
    }
}
```

$\log n$

$\log n$

$n/2$

$O(n \cdot (\log n)^2)$

b)

Leia(n);

x ← 0

Para i ← 1 até n faça n

Para j ← i+1 até n faça n

Para k ← 1 até j-i faça n

x← x + 1

$O(n^3)$

Questão 3

Suponha um algoritmo A e um algoritmo B com funções de complexidade de tempo $a(n) = n^2 - n + 549$ e $b(n) = 49n + 49$, respectivamente. Determine quais são os valores de n pertencentes ao conjunto dos números naturais para os quais A leva menos tempo para executar do que B.

Questão 4

O Casamento de Padrões é um problema clássico em ciência da computação e é aplicado em áreas diversas como pesquisa genética, editoração de textos, buscas na internet, etc. Basicamente, ele consiste em encontrar as ocorrências de um padrão P de tamanho m em um texto T de tamanho n. Por exemplo, no texto T = "PROVA DE AEDSII" o padrão P = "OVA" é encontrado na posição 3 enquanto o padrão P ="OVO" não é encontrado. O algoritmo mais simples para o casamento de padrões é o algoritmo da "Força Bruta", mostrado abaixo. Analise esse algoritmo e responda: Qual é a função de complexidade do número de comparações de caracteres efetuadas no melhor caso e no pior caso. Dê exemplos de entradas que levam a esses dois casos. Explique sua resposta!

3) $n^2 - n + 549 < 49n + 49$

$n^2 - 50n + 500 < 0$

$\Delta = 50^2 - 4 \cdot 500 = 500$

$x \cong \dfrac{50 \pm 22.36}{2} \cong \begin{cases} 36.18 \\ 13.82 \end{cases}$

$14 \leq n \leq 36$

```c
#define MaxTexto 100
#define MaxPadrao 10

/* Pesquisa o padrao P[1..m] no texto T[1..n] */
void ForcaBruta(    char T[MaxTexto], int n,
                    char T[MaxPadrao], int m)
{
    int i,j,k;
    for( i = 0 ; i < n - m + 1 ; i++ )
    {
        k = i;
        j = 0;
        while ( ( j <= m ) && ( T[k] == P[j] ) )
        {
            j = j + 1;
            k = k + 1;
        }
        if (j > m)
        {
            printf("Casamento na posicao %d",i);
            break;
        }
    }
}
```

MELHOR CASO:

PADRÃO NO INÍCIO DO TEXTO:

T = "EDA: MELHOR CADEIRA"

P = "EDA"

$\Theta(m)$

FOR 1 VEZ

WHILE m VEZES

PIOR CASO:

~~PAD~~

TEXTO POSSUI CARACTERES SEMELHANTES AO PADRÃO:

TEXTO: "EEEEE"

PADRÃO: ~~"EEA"~~ "EA"

FOR: ~~[illegible]~~ $n-m+1$

WHILE: m

$O(m.(n-m+1))$

Questão 5

Considere que você tenha um problema para resolver e duas opções de algoritmos. O primeiro algoritmo é quadrático tanto no pior caso quanto no melhor caso. Já o segundo algoritmo, é linear no melhor caso e cúbico no pior caso. Considerando que o melhor caso ocorre 90% das vezes que você executa o programa enquanto o pior caso ocorre apenas 10% das vezes, qual algoritmo você escolheria? Justifique a sua resposta em função do tamanho da entrada.

Questão 6

Perdido em uma terra muito distante, você se encontra em frente a um muro de comprimento infinito para os dois lados (esquerda e direita). Em meio a uma escuridão total, você carrega um lampião que lhe possibilita ver apenas a porção do muro que se encontra exatamente à sua frente (o campo de visão que o lampião lhe proporciona equivale exatamente ao tamanho de um passo seu). Existe uma porta no muro que você deseja atravessar. Supondo que a mesma esteja a n passos de sua posição inicial (não se sabe se à direita ou à esquerda), elabore um algoritmo para caminhar ao longo do muro que encontre a porta em O(n) passos. Considere que n é um valor desconhecido (informação pertencente à instância). Considere que a ação composta por dar um passo e verificar a posição do muro correspondente custa O(1).

(5)

A: $\frac{90n^2}{100} + \frac{10n^2}{100}$

B: $\frac{90}{100}n + \frac{10}{100}n^3$

$A < B$:

$n^2 < \frac{n^3 + 9n}{10}$

$10n^2 - n^3 - 9n < 0$

$n^3 - 10n^2 + 9n > 0$

$n(n^2 - 10n + 9) > 0$

• $n > 0$

• $n^2 - 10n + 9 = 0$

$\Delta = 100 - 4.9 = 64$

$n = \frac{10 \pm 8}{2} = \begin{cases} 9 \\ 1 \end{cases}$

COMO ESTAMOS INTERESSADOS EM $n \in \mathbb{N}$, TEMOS:

$n > 9$

LOGO, TEMOS QUE O PRIMEIRO ALGORITMO É MELHOR QUE O SEGUNDO PARA $n > 9$. PORTANTO, CONSIDERANDO AS PROBABILIDADES DE MELHOR E PIOR CASO, JUNTAMENTE COM O FATO QUE $\forall n, n > 9$ O PRIMEIRO EXECUTA MAIS RÁPIDO, O MESMO É PREFERÍVEL. VEJA QUE A DECISÃO PODERIA SER DIFERENTE SE SOUBESSEMOS A PRIORI QUE A MAIOR PARTE DAS ENTRADAS TEM TAMANHO $n \leq 9$.

6 ENTRADA(n):

1. ANDAR n PASSOS PARA ESQUERDA;	$\Theta(n)$
2. VERIFICAR SE EXISTE PORTA;	$\Theta(1)$
3. SE EXISTE; RETORNA;	$\Theta(1)$
4. SENÃO: ANDA $2n$ PASSOS PARA A DIREITA	$\Theta(n)$
5. RETURN.	$\Theta(1)$

LOGO, NO MELHOR CASO A PORTA ESTÁ A n PASSO À ESQUERDA. $\Omega(n)$

NO PIOR CASO ESTÁ A n PASSOS À DIREITA, ONDE TEMOS $O(n) + O(2n) = O(n)$.

LOGO: $\Theta(n)$.