

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: 18/08/2019

OBS 1: Exercício Individual.

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

NOME: Felipe Marcel de Queiroz Santos

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, ...)

$f(n)$	$O(n)$	Ordem
$5 + 0.001n^3 + 0.025n$	$O(n^3)$	1º
$500n + 100n^{1.5} + n\log_{10}(n)$	$O(n^{1.5})$	5º
$0.3n + 5n^{1.5} + 2.5n^{1.75}$	$O(n^{1.75})$	4º
$n^2\log_2 n + n(\log_2 n)^2$	$O(n^2\log_2 n)$	2º
$n\log_3 n + n\log_2 n$	$O(n\log_2 n)$	6º
$3\log_8 n \log_2 \log_2 \log_2 n$	$O(\log_2 n)$	8º
$100n + 0.01n^2$	$O(n^2)$	3º
$0.01n + 100n^2$	$O(n^2)$	3º
$2n + n^{0.5} + 0.5n^{1.25}$	$O(n^{1.25})$	7º
$0.01n\log_2 n + n(\log_2 n)^2$	$O(n\log_2 n)$	6º

$100n\log_3 n + n^3 + 100n$	$O(n^3)$	1º
$0.003\log_4 n + \log_2 \log_2 n$	$O(\log_2 n)$	8º

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;
        }
    }
}

```

Resposta:

1 – *for* (*mais externo*) \Rightarrow *custo* = $\log_2 n$

2 – *for* (*intermediário*) \Rightarrow *custo* = $\log_2 n$

3 – *for* (*mais interno*) \Rightarrow *custo* = n

Custo total $\Rightarrow O(n \log_2 n \log_2 n)$

b)

Leia(n);

$x \leftarrow 0$

Para $i \leftarrow 1$ até n faça

 Para $j \leftarrow i+1$ até n faça

 Para $k \leftarrow 1$ até $j-i$ faça

$x \leftarrow x + 1$

Resposta:

1 – primeiro laço (mais externo) $\Rightarrow n$

2 – segundo laço (intermediário) $\Rightarrow n - 1$

3 – terceiro laço (mais interno) $\Rightarrow n - 2$

Custo total $\Rightarrow O(n^3)$

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.

Resposta:

$$14 \leq n \leq 36$$

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!

Resposta:

Pior caso, percorrer todo o texto e não encontrar o padrão.

Custo $\Rightarrow O(mn - m^2 + m)$, para casos em que n é muito maior que m, o custo pode ser determinado por $O(mn)$

Melhor caso, quando $m = n$ e m já é o próprio padrão.

Portanto, $(mn - m^2 + m)$, então $O(m)$

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.

Resposta:

Como a análise de algoritmos leva em consideração o pior caso, o custo do algoritmo A para o pior caso é $O(n^2)$, enquanto o custo para o algoritmo B é $O(n^3)$. Entretanto um fator importante a se considerar é que a ocorrência do melhor caso no algoritmo A é muito maior que a ocorrência do pior caso. Partindo dos fatos acima, para entradas com o tamanho de n muito grande, no pior caso, o algoritmo B levará um tempo muito maior para executar o algoritmo, portanto o algoritmo A é mais indicado.

Há ainda, uma outra linha de raciocínio. Se o usuário souber quais são as condições que levam o algoritmo B ao pior caso, é possível fazer um tratamento na utilização dos algoritmos e utilizar ambos. Quando o pior caso for encontrado, utiliza-se o algoritmo A, para todos os outros casos utiliza-se o algoritmo B.

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 $O(1)$.

Resposta:

Escolha um lado.

Dê n passos para o lado escolhido

Se porta está a n passos

fim

se não dê $2n$ passos para o lado contrário