PUC Minas

Lista 4 – Complexidade Assintótica

INFORMAÇÕES DOCENTE						
CURSO: ENGENHARIA DE SOFTWARE	**DISCIPLINA:** FUNDAMENTOS DE PROJETO E ANÁLISE DE ALGORITMOS	**TURNO**	MANHÃ	TARDE	NOITE	**PERÍODO/SALA:** 5º
					x	
PROFESSOR (A): João Paulo Carneiro Aramuni						

Lista 4 - Gabarito

Complexidade Assintótica – Teorema Mestre

1) Dada a seguinte recorrência: $T(n) = 2T(n / 2) + O(n)$
a) Identifique os valores de a, b e $f(n)$.
b) Determine a complexidade assintótica de $T(n)$ usando o Teorema Mestre, identificando o caso aplicado.

2) Dada a seguinte recorrência: $T(n) = 4T(n / 2) + O(n^2)$
a) Identifique os valores de a, b e $f(n)$.
b) Determine a complexidade assintótica de $T(n)$ usando o Teorema Mestre, identificando o caso aplicado.

Respostas:

1) $T(n) = 2T(n / 2) + O(n)$

Passo 1: Identificação dos parâmetros
Comparando com a forma geral $T(n) = aT\ (n / b) + f(n)$, temos:

- $a = 2$: cada instância do problema gera 2 subproblemas de tamanho $n / 2$.
- $b = 2$: o tamanho do problema é reduzido para $n / 2$ a cada passo.
- $f(n) = O(n)$: o custo externo (tempo gasto fora da chamada recursiva) é linear.

Passo 2: Aplicação do Teorema Mestre
Calculamos: $\log_b a = \log_2 2 = 1$

Agora, comparamos $f(n) = O(n)$ com $O(n^{\log_2 2}) = O\ (n^1)$:
$O(n)$ vs $O(n^1)$
Como os dois termos são iguais, aplicamos o Caso 2 do Teorema Mestre, que afirma:
Se $f(n) = \Theta(n^{\log_b a})$, então $T(n) = \Theta(n^{\log_b a} \log n)$.

Como $n^{\log_2 2} = n$, a complexidade final é:
$T(n) = \Theta(n \log n)$

Passo 3: Conclusão e exemplo de algoritmo

Essa recorrência descreve o comportamento do Merge Sort, que divide o problema em duas partes iguais (daí $a = 2$, $b = 2$) e gasta tempo linear para mesclar os subproblemas.

Resumo:
- Parâmetros: $a = 2$, $b = 2$, $f(n) = O(n)$.
- Caso do Teorema Mestre aplicado: Caso 2.
- Complexidade final: $\Theta\ (n \log n)$.
- Algoritmo correspondente: Merge Sort.

2) $T(n) = 4T(n / 2) + O(n^2)$

Passo 1: Identificação dos parâmetros
Comparando com a forma geral $T(n) = aT\ (n / b) + f(n)$, temos:

- $a = 4$: o problema é dividido em 4 subproblemas.
- $b = 2$: cada subproblema tem tamanho $n / 2$.
- $f(n) = O(n^2)$: o custo externo é quadrático.

Passo 2: Aplicação do Teorema Mestre
Calculamos: $\log_b a = \log_2 4 = 2$

Agora, comparamos $f(n) = O(n^2)$ com $O(n^{\log_2 4}) = O(n^2)$:
$O(n^2)$ vs $O(n^2)$

Como os dois termos são iguais, novamente aplicamos o Caso 2 do Teorema Mestre:
$T(n) = \Theta(n^2 \log n)$

Passo 3: Conclusão e exemplo de algoritmo

Essa recorrência aparece em algoritmos de multiplicação de matrizes por divisão e conquista, onde o problema é subdividido em 4 submatrizes, e a combinação dos resultados exige um tempo quadrático.

PUC Minas

Resumo:

- Parâmetros: a = 4, b = 2, $f(n)$ = $O(n^2)$.
- Caso do Teorema Mestre aplicado: Caso 2.
- Complexidade final: $\Theta(n^2 \log n)$.
- Algoritmo correspondente: Multiplicação de matrizes ingênua por divisão e conquista.
