Projeto e Análise de Algoritmos

Resolvendo Problemas de Algoritmos de Ordenação

Matheus Gabriel

Agosto de 2024

1 Primeiro problema

1.1 Enunciado

1.1.1 Contexto

Professor Rubião afirma ter encontrado uma versão assintoticamente melhor que a versão original. Na versão de Rubião *merge sort* **divide** o vetor em 3 partes e chama recursivamente merge sort para cada uma destas partes. Finalmente é chamado uma versão modificada do subalgoritmo merge para intercalar o resultado destas 3 chamadas.

Sua tarefa é verificar se Prof Rubião realmente melhorou merge sort assintoticamente.

1.1.2 Exemplo

msort	1	2	3
msort	1	–	–
msort	–	2	–
msort	–	–	3
merge'	–	–	–

1.2 Como fazer

Para resolver, você deve:

1. Achar a fórmula aberta para o algorítmo do Prof. Rubião
2. Encontrar a fórmula fechada desta fórmula aberta.
3. Comparar este resultado com $T(n) = \Theta(n \lg n)$ que é a fórmula fechada do merge sort original.

1.3 Resolução

1.3.1 Achando a fórmula aberta

Note a análise na seção exemplo, lembre-se que $T(n)$ é o **T**empo necessário para resolver o algorítmo de tamanho **n**

$$T(1) = 1$$
$$T(n) = 3 \cdot T(\frac{n}{3}) + n$$

1.3.2 Encontrando a fórmula fechada

$$\begin{aligned}
T(n) &= 3T\left(\frac{n}{3}\right) + n \\
\text{Mas } T\left(\frac{n}{3^2}\right) &+ \frac{n}{3} \\
\therefore T(n) &= 3\left(3T\left(\frac{n}{3^2}\right) + \frac{n}{3}\right) + n \\
T(n) &= 3^2T\left(\frac{n}{3}\right) + n + n \\
\text{Mas } T\left(\frac{n}{3^2}\right) &= 3 \cdot T\left(\frac{n}{3^3}\right) + \frac{n}{3^2} \\
\therefore T(n) &= 3^2\left(3T\left(\frac{n}{3^3}\right) + \frac{n}{3^2}\right) + n + n \\
T(n) &= 3^3T\left(\frac{n}{3^3}\right) + n + n + n
\end{aligned}$$

Generalizando, temos:

$$T(n) = 3^iT\left(\frac{n}{3^i}\right) + \sum_{k=1}^{i} n$$

e vai parar quando $\frac{n}{3^i} = 1 \longrightarrow \boxed{i = \log_3 n}$

$$\begin{aligned}
\therefore T(n) &= 3^{\log_3 n}[T(1) \to 1] + \sum_{k=1}^{\log_3 n} n \\
T(n) &= n^{\log_3 3} + n\log_3 n = n + n\log_3 n \\
\therefore T(n) &= \Theta(n\log_3 n) = \boxed{\Theta(n\lg n)}
\end{aligned}$$

1.3.3 Comparando com merge sort normal

Não houve melhoria *assintótica*, portanto não vale a pena atualizar o merge sort.

2 Segundo problema

2.1 Enunciado

Encontre a fórmula fechada para:

$$T(n) = 2T\left(\frac{n}{2}\right) + n^2$$

2.2 Resolução

Consulte a fórmula: $\sum_{k=0}^{n} c^k = \frac{c^{n+1}-1}{c-1}$

$$\begin{aligned}
T(n) &= 2T\left(\frac{n}{2}\right) + n^2 \\
\text{Mas } T\left(\frac{n}{2}\right) &= 2T\left(\frac{n}{2^2}\right) + \frac{n^2}{2^2} \\
\therefore T(n) &= 2\left(2T\left(\frac{n}{2^2}\right) + \frac{n^2}{2^2}\right) + n^2 \\
T(n) &= 2^2T\left(\frac{n}{2^2}\right) + \frac{n^2}{2} + n^2 \\
\text{Mas } T\left(\frac{n}{2^2}\right) &= 2T\left(\frac{n}{2^3}\right) + \frac{n^2}{2^4} \\
\therefore T(n) &= 2^2\left(2T\left(\frac{n}{2^3}\right) + \frac{n^2}{2} + n^2\right) \\
T(n) &= 2^3T\left(\frac{n}{2^3}\right) + \frac{n^2}{2^2} + \frac{n^2}{2^1} + \frac{n^2}{2^0}
\end{aligned}$$

Generalizando, temos:

$$T(n) = 2^i T\left(\frac{n}{2^i}\right) + n^2 \sum_{k=0}^{i-1} \left(\frac{1}{2}\right)^k$$

$$\text{e vai parar quando } \frac{n}{2^i} = 1$$

$$\text{ou } \boxed{i = \log_2 n = \lg n}$$

$$\therefore T(n) = 2^{\log_2 n}[T(1) \to 1] + n^2 \sum_{k=0}^{\log_2 n=1} \left(\frac{1}{2}\right)^k$$

$$T(n) = n^{\log_2 2} + n^2 \left(\frac{\left(\frac{1}{2}\right)^{\log_2 n} - 1}{\frac{1}{2} - 1}\right) =$$

$$= n + n^2 \left(\frac{n^{\log_2 \frac{1}{2}} - 1}{-\frac{1}{2}}\right) = n + n^2 \left(\frac{n^{-1} - 1}{-\frac{1}{2}}\right)$$

$$\ldots$$

$$\therefore \boxed{T(n) = \Theta(n^2)}$$

Faltou completar os cálculos na aula.