



Análise de Algoritmos - Introdução

Tarefa - Análise de Algoritmos - Introdução

1 Funções e tempos de execução

- (Retirado de [Thomas H. Cormen, 2022]) Suponha que estamos comparando os tempos de execução dos algoritmos de ordenação *insertion sort* e *merge sort*. Para entradas de tamanho n , o algoritmo *insertion sort* utiliza $8n^2$ passos para executar seu trabalho enquanto o algoritmo *merge sort* utiliza $64n \log_2[n]$ passos. Para quais valores de n o algoritmo *insertion sort* é mais rápido que o *merge sort*. Justifique sua resposta e descreva como n foi calculado.
- (Retirado de [Thomas H. Cormen, 2022]) Qual é o menor número n tal que um algoritmo cujo tempo de execução é $100n^2$ é mais rápido que um algoritmo cujo tempo de execução é 2^n . Justifique sua resposta e descreva como n foi calculado.
- (Retirado de [Thomas H. Cormen, 2022]) Assumindo que um programa utiliza $f[n]$ microsegundos ($1s = 10^6 \mu s$) para resolver um problema com uma entrada de tamanho n , determine a maior entrada n que o programa poderá resolver para cada um dos tempos t e funções $f[n]$ presentes na tabela abaixo.

	1 segundo	1 minuto	1 hora	1 dia	1 mês	1 ano	1 século
$\log_2[n]$							
$[n]^{\frac{1}{2}}$							
$n \log_2[n]$							
n^2							
n^3							
2^n							
$n!$							

Descreva o método utilizado para preencher os campos da tabela.

- (Retirado de [Thomas H. Cormen, 2022]) Escreva o algoritmo *insertion sort* para ordenar o array em ordem monotônica decrescente.
- (Retirado de [Thomas H. Cormen, 2022]) Considere o seguinte problema de busca:

Entrada: Uma sequência de números (a_n) armazenada em um array $A[1 : n]$ e um valor x .

Saída: Um índice i tal que $x = A[i]$ ou o valor especial *null* se x não aparece em (a_n) .

Escreva um pseudocódigo para a busca linear, uma busca que vai da primeira até a última posição da sequência buscando pelo valor x . Prove que seu algoritmo está correto usando

um invariante de loop. Assegure-se de que seu invariante cumpre as três propriedades apresentadas em [Thomas H. Cormen, 2022].

6. (Retirado de [Thomas H. Cormen, 2022] e modificado) Considere o problema de adicionar dois inteiros binários a e b , ambos de tamanho n . Os números a e b estarão armazenados nos arrays $A[0 : n - 1]$ e $B[0 : n - 1]$ nos quais cada posição somente será 0 ou 1. O valor inteiro de a é obtido por

$$a = \sum_{i=0}^{n-1} A[i]2^i$$

O valor de b é obtido de maneira similar. A soma dos inteiros binários a e b devem ser armazenadas em um inteiro c que será representado por um array $C[0 : n]$

Escreva um pseudocódigo para a soma binária dos números a e b que retorne um número c representado por um array binário $C[0 : n - 1]$ e um valor binário chamado *vaium* que deverá ser 1 quando a soma de a e b excede a “precisão” do array.

Referências

- [Thomas H. Cormen, 2022] Thomas H. Cormen, Charles E. Leiserson, R. L. R. C. S. (2022). *Introduction to Algorithms*. The MIT Press, 4 edition.