

2ª Lista de Exercícios

AEDS III 2019

Prof. Daniel Conrado

Exercício 1. (adaptado de CORMEN, 2009) Faça um teste de mesa dos algoritmos merge sort e comb sort para a seguinte instância do problema:

$$A = [3, 41, 52, 26, 38, 57, 9, 49]$$

Utilize como modelo de teste de mesa a planilha disponível [aqui](#).

Exercício 2. Dado o seguinte problema computacional:

Entrada: dois valores inteiros positivos a e n

Saída: a^n

- Escreva um programa em C++ que soluciona esse problema utilizando a abordagem força bruta.
- Escreva um programa em C++ que soluciona esse problema utilizando a abordagem dividir-e-conquistar (ou decrementar-e-conquistar).

Exercício 3. Considere o problema de adicionar dois inteiros de n bits, armazenados em dois vetores de n elementos chamados A e B . A soma dos dois inteiros deve ser armazenada em em formato binário um terceiro vetor C de $n+1$ elementos. Podemos descrever formalmente esse problema da seguinte maneira:

Entrada: duas sequências de n elementos $A = (a_1, a_2, \dots, a_n)$ e $B = (b_1, b_2, \dots, b_n)$ tal que $\forall a_i \in \{0, 1\}$ e $\forall b_i \in \{0, 1\}, 1 \leq i \leq n$, cada uma representando um número inteiro armazenado em formato binário.

Saída: uma sequência $C = (c_1, c_2, \dots, c_{n+1})$ tal que $C' = A' + B'$ em que C' , A' e B' são inteiros representados por A , B e C .

Para facilitar o entendimento, considere a seguinte instância desse problema e sua solução:

Entrada:

$A = [1, 1, 0, 1]$ (número 13 em binário)

$B = [1, 0, 1, 0]$ (número 10 em binário)

Saída:

$C = [1, 0, 1, 1, 1]$ (número 23 em binário), pois $1101 + 1010 = 10111$.

Agora, escreva um programa em C++ para solucionar esse problema e realize um teste de mesa para a instância $A=[1, 1, 1]$, $B=[1, 0, 1]$.

Exercício 4. O algoritmo Insertion Sort *insere* uma chave desordenada na posição correta da parte da sequência que já está ordenada, começando da segunda chave. Quando a chave a inserir é a segunda, a parte da sequência já ordenada é composta apenas do primeiro elemento. Quando a chave a inserir é a terceira, a parte da sequência já ordenada é composta do primeiro e segundo elementos. Quando a chave a inserir é a quarta, a parte da sequência já ordenada é composta do primeiro, segundo e terceiro elementos, já ordenados, e assim por diante. Elabore uma versão do Insertion Sort que, ao invés de começar da segunda chave, comece da penúltima. Faça um teste de mesa com a sequência $S = [30, 16, 32, 29, 28, 16]$.

Exercício 5. Embora MergeSort e InsertionSort executem, respectivamente, em $O(n \lg n)$ e $O(n^2)$ no pior caso, os fatores constantes do InsertionSort podem torná-lo mais rápido na prática para problemas pequenos em muitos computadores. Então, faz sentido diminuir a “árvore” de recursão do MergeSort utilizando o InsertionSort quando os subproblemas forem suficientemente pequenos. Agora, responda:

- a) Escreva o algoritmo MergeSort com uma pequena modificação: para subproblemas com menos de k elementos, esses devem ser resolvidos com InsertionSort ao invés do MergeSort e depois combinados utilizando o mecanismo padrão de combinação do MergeSort.
- b) Faça um teste de mesa para a instância $[11, 10, 2, 13, 8, 0, 6, 6, 4, 1]$ e $k = 3$.