

Lista de Exercícios 5: Algoritmos de Ordenação II

Professor: Moacir Pereira Ponti Jr.

PAE(s): Pâmela/Paulo

1. Quantas comparações são necessárias para realizar um “merge” de dois conjuntos ordenados com n elementos cada?
2. Escreva um algoritmo para a rotina $merge(x, i, j, k)$, que presuma que $x[i]$ até $x[j]$ e $x[j + 1]$ até $x[k]$ estão classificados e intercale os dois em $x[i]$ até $x[k]$.
3. Faça um teste de mesa com os método de ordenação *heapsort*, *mergesort* e *shellsort*, utilizando as seguintes sequências de dados de entrada:
 - (i) $S_1 = \{2, 4, 6, 8, 10, 12\}$
 - (ii) $S_2 = \{11, 9, 7, 5, 3, 1\}$
 - (iii) $S_3 = \{5, 7, 2, 8, 1, 6\}$
 - (iv) $S_4 = \{2, 4, 6, 8, 10, 12, 11, 9, 7, 5, 3, 1\}$
 - (v) $S_5 = \{2, 4, 6, 8, 10, 12, 1, 3, 5, 7, 9, 11\}$
 - (vi) $S_6 = \{8, 9, 7, 9, 3, 2, 3, 8, 4, 6\}$
 - (vii) $S_7 = \{89, 79, 32, 38, 46, 26, 43, 38, 32, 79\}$

Diga, ainda, quais desses algoritmos são estáveis.

4. Uma **ordenação por contagem** (*countsort*) de um vetor x de tamanho n é executada da seguinte forma: declare um vetor `count` e defina `count[i]` como o número de elementos menores que $x[i]$. Em seguida, coloque $x[i]$ na posição `count[i]` de um vetor de saída (leve em consideração a possibilidade de elementos repetidos). Escreva uma função para ordenar um vetor x de tamanho n usando esse método.
5. Presuma que um vetor contém inteiros entre a e b , inclusive, com vários números repetidos diversas vezes. Uma **ordenação por distribuição** (*bucketsort*) ocorre da seguinte maneira: declare um vetor `number` de tamanho $b - a + 1$, defina `number[i - a]` como o número de vezes que o inteiro i aparece no vetor e , em seguida, redefina os valores no vetor concomitantemente. Escreva uma função para ordenar um vetor x de tamanho n com esse método.
6. Escreva uma versão recursiva do algoritmo *bubblesort*. Nessa versão, o principal comando de repetição deve ser substituído por uma chamada recursiva, controlada por um contador.

Referências

- [1] Parte deste material foi adaptada das listas de exercícios do Prof. Luis Gustavo Nonato, ICMC/USP.
- [2] Preiss, B. R., *Data Structures and Algorithms with Object-Oriented Design Patterns in C++*. 1997.
- [3] Parte deste material foi adaptada das listas de exercícios do Prof. João Luís Garcia Rosa, ICMC/USP.