13ª Lista de Exercícios

Estrutura de Dados

Prof. Hamilton José Brumatto

Algoritmos de Ordenação em Tempo Linear

1. Considere a seguinte sequência de números: 5 3 7 5 9 4 8 2 7 6. Simule a ordenação utilizando o algoritmo Counting Sort.
2. Considere a segunte sequência de números: 194 439 34 952 933 234 356 342 669 463 13 555. Simule a ordenação utilizando o algoritmo Radix-Sort.
3. Implemente um programa para realizar a ordenação de um conjunto de números utilizando o algoritmo Radix-Sort, e teste este programa no conjunto de números do item anterior.
4. Na lista anterior você testou alguns algoritmos, compare o tempo do Radix-Sort com aqueles algoritmos na ordenação.
5. Um detalhe importante do Radix sort é que ele é aplicável em números que possuem um limite na quantidade de dígitos. Ora, o inteiro no computador é exatamente isto, um número que possui uma quantidade limitada de dígitos (*unsigned short* tem 8 bits, por exemplo). Uma idéia para ordenar inteiros positivos é utilizar Radix Sort com dígitos binários. A vantagem disto é que aliando o Counting Sort neste caso, o vetor de contagem: C terá somente duas posições: 0 e 1. Implemente uma função que receba como parâmetros um vetor de inteiros positivos, um inteiro indicando a quantidade de elementos neste vetor e um inteiro que indique o tamanho máximo de dígitos binários que estes inteiros atendem. Utilize, nesta função, o algoritmo do Radix Sort para ordenar estes valores.
6. O algoritmo Bucket Sort trabalha com valores de [0..1). Considere a sequência da questão 2, podemos dividir os valores por 1000 e obter um conjunto compatível com o algoritmo. Faça uma simulação de ordenação para esta sequência. Para apresentar o resultado, retome o produto por 1000 para obter os valores originais.
7. Implemente um algoritmo para o Bucket Sort. E processe a sequência anterior usando este algoritmo.

.