unioeste
Universidade Estadual do Oeste do Paraná

Exercícios de Algoritmos e Estrutura de Dados
Ciência da Computação
campus Foz do Iguaçu

Data: Junho/2016
Prof. Rômulo Silva

Tópicos: **Busca Binária, Algoritmos de Ordenação**

1. Escreva uma função que recebba um vetor estritamente crescente $vet[0..n-1]$ de números inteiros e devolva um índice i entre 0 e $n-1$ tal que $vet[i] = i$; se tal i não existe, a função deve retornar -1. O seu algoritmo não deve fazer mais que $O(lg\, n)$ comparações.

2. Escreva um função de complexidade $O(lg\, n)$ que receba parâmetros inteiros k e n e retorne o valor de k^n.

3. Escreva uma versão recursiva do algoritmo de ordenação por seleção (*Selection Sort*).

4. Escreva uma função que ordene uma lista encadeada baseado no algoritmo de ordenação por inserção. Faça 2 versões: uma para lista sem cabeça e outra para lista com cabeça.

5. Considerando as implementações dos algoritmos de ordenação mostrados em sala de aula (*Selection Sort*, *Insertion Sort*, *Mergesort*, *Bubble Sort* e *Quicksort*), quais deles são estáveis e quais não são?

6. Pesquise sobre o algoritmo de ordenação *Shellsort*. Qual a complexidade de tempo desse algoritmo? É um método estável?

7. O problema da ordenação parcial pode ser formulado da seguinte maneira: dado um vetor $vet[0..n-1]$ desordenado e um inteiro $k \leq n$, obter os k primeiros elementos de $vet[0..n-1]$ ordenado. Observe que $k = 1$ corresponde a obter o menor elemento, enquanto $k = n$ corresponde ao problema clássico da ordenação. Implemente uma função que receba como parâmetros $vet[0..n-1]$, n e k, e faça a ordenação parcial baseada no algoritmo de ordenação por seleção.