

# DCA0214.1 - LABORATÓRIO DE ESTRUTURAS DE DADOS

## Aula 3: ordenação em listas sequenciais (vetores)

Prof. Felipe Fernandes

16 Agosto de 2019

1. Implemente os seguintes algoritmos de ordenação:
  - (a) InsertionSort
  - (b) SelectionSort
  - (c) BubbleSort
  - (d) MergeSort
  - (e) QuickSort
  - (f) CountingSort
  - (g) RadixSort
  - (h) BucketSort
2. Crie um algoritmo chamado **quickfind** baseado no quicksort para que, em vez de ordenar um vetor de números inteiros, ele nos retorne o  $k$ -ésimo menor elemento desse vetor. O procedimento **quickfind** deve ter a seguinte interface: **quickfind**( $V, p, r, k$ ), onde  $V$  é um vetor de inteiros,  $p$  é o menor índice de  $V$ ,  $r$  o maior índice, e  $1 \leq k \leq n$  um inteiro positivo. Por exemplo: para  $V = [7, 1, 3, 10, 17, 2, 21, 9]$ , a chamada **quickfind**( $V, 1, 8, 5$ ) deverá retornar o número 9 (quinto menor elemento). Escreva o procedimento **quickfind** recursivamente (pseudo-código), modificando o procedimento quicksort visto em aula. Obs.: Você não deve simplesmente ordenar todo o vetor e depois tomar o  $k$ -ésimo elemento.
3. Implemente o algoritmo de ordenação por inserção de modo recursivo da seguinte maneira: para ordenar o vetor  $V[1...n]$ , ordene recursivamente  $V[1...n-1]$  e então insira o elemento  $V[n]$  no vetor ordenado  $V[1...n-1]$ . Compare experimentalmente o desempenho das versões recursivas e iterativas.
4. Seja  $A$  um vetor de  $n$  números inteiros distintos. Se  $i < j$  e  $A[i] > A[j]$  então o par  $(i, j)$  é chamado uma inversão  $A$ . Escreva um algoritmo, cujo pior caso seja  $\Theta(n \log n)$ , que calcule a quantidade de inversões em  $A$ .

5. Escreva uma função que recebe dois argumentos: (1) um vetor  $V[0 \dots n-1]$  com  $n$  inteiros, (2) um inteiro  $x$ ; e determina se existem dois elementos em  $V$  cuja soma é exatamente  $x$ . A complexidade do seu algoritmo deve ser  $\Theta(n \log n)$ .