

---

7ª Lista de Exercícios – Ordenação

Data Entrega: vide site

Enviar e-mail:

- Assunto: Lista7
- Anexo: Um único arquivo .ZIP com todos os arquivos .doc, .png, .jpg, .cpp, .h, .dev, .sln, etc.

**OBS:** e-mails fora deste padrão serão desconsiderados

**OBS 2:** quem preferir pode entregar os passo-a-passo desta lista em papel

---

1. Faça o passo a passo da ordenação dos vetores abaixo utilizando as seguintes ordenações:
  - a. BubbleSort
  - b. InsertionSort
  - c. SelectionSort
  - d. ShellSort (utilizando potências de 2 como incrementos)
  - e. MergeSort
  - f. QuickSort
  - g. HeapSort

Vetores:

- 1) 15 3 4 19 26 7 9 11
- 2) 20 19 17 15 14 12 11 9 7 4 2 1

2. Implemente o ShellSort de forma parametrizável, isto é, o usuário deve passar para o ShellSort duas funções de ordenação como parâmetros, a primeira função de ordenação será utilizada quando  $H \neq 1$ , e a segunda função de ordenação será utilizada quando  $H = 1$ . As funções de ordenação que o usuário pode passar como parâmetro podem ser: InsertionSort, SelectionSort e BubbleSort. As assinaturas destas três funções devem receber o H como parâmetro.
3. Implemente o BubbleSort de forma a ele mudar de direção a cada passo. Isto é, o BubbleSort “borbulha” nas duas direções. No primeiro passo a direção é da esquerda para a direita (então o maior elemento irá ser posicionado na última posição do vetor), no segundo passo a direção é da direita para a esquerda (então o menor elemento irá ser posicionado na primeira posição do vetor), e assim por diante.
4. Crie uma função que receba como parâmetro uma função de ordenação e retorne um booleano informando se esta função de ordenação é ou não estável.