

## Trabalho V1

**Pontuação:** 10.0 pontos

**Data da Entrega:** 04/ 10/ 2020 até as 22h.

1. Deverão ser gerados vários vetores para ordenação, como exemplo seguem alguns parâmetros que poderão ser utilizados:
  - a. **start:** recebe a quantidade de itens que o vetor iniciará a ordenação. Por exemplo: caso o valor seja 10.000 o primeiro vetor ordenado terá essa quantidade de itens.
  - b. **stop:** recebe a quantidade máxima de itens do último vetor a ser ordenado
  - c. **step:** é o step de crescimento dos vetores, ou seja, a razão de crescimento. Por exemplo, se você escolher 1.000 a cada iteração o vetor irá crescer 1000 posições, assim, o tamanho dos vetores seriam os seguintes (por iteração): 1000, 2000, 3000, etc.
  - d. **mostra\_vetor (opcional):** a cada iteração irá mostrar o vetor ordenado  
- **True** mostra o vetor e **False** não mostra
  - e. **mostra\_tempo (opcional):** a cada iteração irá mostrar o tempo para a ordenação do vetor - **True** mostra o tempo e **False** não mostra.
2. O Array gerado no passo 1 deverá ser ordenado pelos seguintes algoritmos
  - a. Insertion Sort
  - b. Merge Sort
  - c. Radix Sort
3. Para cada algoritmo do passo 2.a, 2.b e 2.c deverão ser medidos os tempos de execução da ordenação.
4. Criar um gráfico comparando todos os 3 algoritmos. O gráfico deverá conter no eixo x a quantidade de elementos ordenados e no eixo y o tempo utilizado para a ordenação.
5. Deverá ter um gráfico exclusivo comparando o merge sort com o Radix sort.
6. Entregáveis
  - a. O Código dos algoritmos radix sort, merge sort e insertion sort
  - b. Um relatório com no máximo 3 slides com os resultados obtidos.
    - i. Nos slides deverá conter os dois gráficos solicitados no item 4 e 5.
    - ii. Uma conclusão do aluno sobre o desempenho dos 3 algoritmos.