



Campus de Cascavel  
Centro de Ciências Exatas e Tecnológicas - CCET  
Curso de Ciência da Computação  
Disciplina: Algoritmos  
Professor: Josué Castro

## Trabalho para a 3<sup>a</sup> Avaliação 2025 (30 pontos)

### Instruções:

1. Este trabalho vale **30 pontos para a 3<sup>a</sup> Avaliação.**
  2. Este trabalho pode ser realizado em **equipes de dois ou três alunos. Em nenhuma hipótese serão aceitos trabalhos individuais ou com mais de três alunos.**
  3. Este trabalho refere-se apenas às **Unidades 4-8** do conteúdo programático, ficando portanto proibida a utilização de outros recursos não cobertos nestas referidas unidades.
  4. **Data de entrega: Até as 23h 59min do dia 15/12/2025**, pelo sistema Teams da Unioeste. **Em nenhuma hipótese será aceita a entrega por outros meios.**
- 

**Tema:** Análise empírica de algoritmos de ordenação

### Descrição:

Deve ser realizada uma análise empírica dos seguintes algoritmos de ordenação listados abaixo:

1. Ordenação pelo Método da Bolha (*Bubblesort*)
2. Ordenação Oscilante (*Shakersort*)
3. Ordenação por Seleção (*Selectionsort*)
4. Ordenação por Inserção (*Insertionsort*)
5. Ordenação pelo método Shell (*Shellsort*)
6. Ordenação por intercalação (*Mergesort*)
7. Ordenação Rápida (*Quicksort*)

### Método:

Cada algoritmo deve ser submetido a várias baterias de teste, aplicadas sobre dois conjuntos de teste fornecidos junto com esta descrição:

#### Conjunto 1:

- Arquivo: **numbers.txt**
- Conteúdo: uma lista não ordenada de números, com 10.000.000 valores inteiros de tipo *unsigned long long*.

#### Conjunto 2:

- Arquivo: **words.txt**
- Conteúdo: uma lista não ordenada de palavras, com 261.788 palavras da língua portuguesa. O comprimento (número de caracteres) da maior palavra da língua portuguesa é 46 caracteres.

### Método:

Aplicar cada algoritmo de ordenação a cada conjunto de entrada, começando com 1000 elementos, e registrando as seguintes informações:

- a) tempo gasto na ordenação nos três casos: melhor caso (conjunto já ordenado), caso médio (desordem aleatória) e pior caso (pegue o conjunto ordenado em ordem crescente e reordene-os em ordem decrescente.)
- b) número de comparações realizadas durante o processo de ordenação nos três casos
- c) número de trocas realizadas durante o processo de ordenação nos três casos

Repita o processo, ampliando o tamanho do conjunto de 1000 em 1000 elementos, até o final do conjunto total de entrada de dados.

Após coletar todos os dados, transporte os dados coletados para uma planilha de dados e plote os gráficos obtidos pelo processo de ordenação para cada informação coletada, e compare-os com as suas respectivas funções limites (p. ex.  $X^2$  para o algoritmo *Bubblesort*, ver *slides* da unidade 8), e para as funções limites para seus respectivos números de comparações e de trocas.

### **Comparação e *ranking* dos algoritmos:**

Após a execução de todos os algoritmos, faça sua análise dos algoritmos, observando todas as informações obtidas através das execuções, para os dois conjuntos. Faça tabelas para comparar os tempos, os números de comparações e de trocas, para os dois conjuntos de dados (*numbers* e *words*). Após isso, construa um *ranking* dos algoritmos, do melhor para o pior, justificando as suas análises com base nos experimentos realizados. Analise também o comportamento do algoritmo (comportamento natural ou não).

### **Como utilizar as entradas de dados:**

Escreva seus programas normalmente, como se estivesse lendo os dados via console. Aloque o vetor a ser ordenado com o tamanho adequado, e então leia os dados para o vetor normalmente usando a função *scanf()* até o limite desejado em cada execução (1000, 2000, 3000, ...).

Compile o programa normalmente. Para executar, utilize o console Linux e execute o programa da seguinte forma:

```
$ ./a.out < numbers.txt > numbers_bolha.out
```

O arquivo *numbers.txt* contém os dados desordenados (entrada), e o arquivo *numbers\_bolha.out* deverá conter o resultado da ordenação (saída) e também as informações obtidas no processo, que devem ser escritas no arquivo após a escrita do vetor.

As mesmas instruções acima valem para o arquivo *words.txt*.

### **O que deve ser entregue:**

As equipes devem elaborar um arquivo seguindo o modelo descrito no arquivo *sbc\_template.doc*. Neste arquivo estão descritas todas as seções que devem estar contidas no artigo e o que deve estar contido nelas.

**Obs: Dúvidas sobre o trabalho podem ser tiradas através do Teams institucional.**

Bom trabalho