Universidade Federal do Paraná
Setor de Ciências Exatas / Departamento de Informática – DInf
Algoritmos e Estruturas de Dados II – CI056 / 1º semestre de 2017
Professor: André Gregio & David Menotti

# Trabalho Prático 2

### Algoritmos de Ordenação por Comparação

# Análise de Desempenho de Algoritmos e suas Variações e Otimizações

Avaliação: 15 pontos (15% do total da disciplina)
Data de divulgação: 05/06/2017
Data de entrega: 27/06/2017

## Objetivos

O objetivo deste trabalho é analisar algoritmos de ordenação por comparação e suas variações e otimizações.

## Descrição

A análise dos algoritmos será dividida em duas partes.

Na primeira, a análise abrange os algoritmos de ordenação simples (de ordem de complexidade $O(n^2)$, i.e., `BubbleSort`, `SelectionSort`, `InsertionSort` e algumas variações). Você deverá implementar versões *clássicas* dos três algoritmos citados e implementar variações desses algoritmos de forma que por meio de experimentos você possa analisar se o custo de realizar as operações abaixo traz algum benefício:

- Inserir verificação de ordenação (se houve troca ou não) no algoritmo `BubbleSort`;
- Usar o algoritmo `InsertionSort` com elemento sentinela;
- Inserir uma verificação ($min == i$) para evitar trocas desnecessárias no método `SelectionSort`.

A segunda parte refere-se aos algoritmos de ordenação por comparação ditos eficientes, como `MergeSort`, `HeapSort` e `QuickSort`, que têm complexidade de tempo de $O(n \log n)$. São fornecidas implementações convencionais e versões otimizadas. Aproveite os algoritmos implementados por você e verifique:

- Até que tamanho de entrada vale a pena usar algoritmos $O(n^2)$ com relação à algoritmos $O(n \log n)$?

## Como gerar os arranjos

Considere arranjos (vetores) com diferentes quantidades de elementos (10, 100, 1.000, 10.000, 100.000, 1.000.000). Considere também arranjos sem valores repetidos. Considere ainda que todos os elementos dos arranjos correspondem a valores inteiros e, para gerar os arranjos iniciais, utilize: (i) arranjos ordenados, (ii) inversamente ordenados, (iii) quase ordenados e (iv) aleatórios.

## O que analisar

A análise deve ser feita sobre o número de *comparações*, *atribuições* e *tempo de execução* dos algoritmos. Procure organizar os dados coletados de modo compreensível em tabelas e construa gráficos a partir desses dados. Discuta os dados obtidos presentes nas tabelas e gráficos. Grande parte da avaliação será realizada sobre a análise dos resultados, ou seja, sobre o que você discutir/dissertar.

## As implementações dos algoritmos

Para este trabalho, estão disponíveis (no site da disciplina) implementações de algoritmos para a análise solicitada na segunda parte. A compilação, interpretação e uso deste código constitui parte da avaliação deste trabalho prático. A implementação dos outros algoritmos a se utilizar na primeira parte fica por sua conta. Você deverá entender o padrão de programação do código fornecido e implementar os novos algoritmos seguindo esse mesmo padrão. Salienta-se que o uso do programa pode ser feito via linha de comando (*prompt*) ou dentro do ambiente de programação que você usa. Todavia, antes, você deverá compilar o código para gerar o arquivo executável e configurar o arquivo `in.txt` para realizar seus testes. Também é fornecido um arquivo exemplo de configuração—**apenas ilustrativo**— que não deve ser usado como referência para seus testes. O resultado da execução do programa, ou seja os dados para você analisar, são gravados em `log.txt`.

## O que deve ser entregue

- Código-fonte dos programas em C (bem indentado e comentado).
- Documentação do trabalho, limitada a 20 páginas incluindo capa e sumário.

Entre outros detalhes, a **documentação** deve conter:

1. ***Introdução***: descrição dos problemas a serem resolvidos.
2. ***Implementação***: descrição das implementações. Deve ser detalhada a estrutura de dados utilizada (de preferência com diagramas e/ou figuras ilustrativos), o funcionamento das principais funções e procedimentos utilizados, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. **Muito importante**: os códigos utilizados nas implementações devem ser inseridos na documentação de maneira organizada.

3. ***Listagem de testes executados***: tabular, construir gráficos a partir dos dados gerados pelo programa e analisar densamente os resultados.

4. ***Conclusão***: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.

5. ***Bibliografia***: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sítio da Internet se for o caso. Uma referência bibliográfica deve ser mencionada (citada) no texto no local em que é utilizada.

6. ***Em LaTeX***: Caso o trabalho não seja elaborado/escrito em LaTeX, perde-se **um ponto**. Veja modelo de como fazer o trabalho em LaTeX:
`http://www.inf.ufpr.br/{gregio,menotti}/ci056-171/tps/modelo.zip`

7. **Impressão**: Caso o trabalho não seja impresso usando modo frente-verso, perde-se **um ponto** (**APENAS PARA ALUNOS DO PROF. MENOTTI**).

8. ***Formato final***: mandatoriamente em PDF (`http://www.pdf995.com/`).

# Como deve ser feita a entrega

A entrega DEVE ser feita via Moodle (`http://moodle.c3sl.ufpr.br/`) na forma de um **único** arquivo compactado (.zip, .tar.gz, .tgz, .rar, etc), contendo o código fonte, arquivos diversos e a documentação. Também deve ser entregue a documentação impressa na próxima aula teórica após a data de entrega do trabalho (**APENAS PARA ALUNOS DO PROF. MENOTTI**).

# Comentários Gerais

- Comece a fazer este trabalho logo, enquanto o prazo para terminá-lo está tão longe quanto jamais poderá estar;
- Clareza, indentação e comentários no programa também serão alvos de avaliação.
- O trabalho é individual (grupo de UM aluno).
- Trabalhos plagiados (e FONTE) terão nota zero. Além disso, os infratores terão a maior nota de provas teóricos levada à zero, como forma de punição e coibição ao plágio acadêmico;
- Trabalhos entregues em atraso serão aceitos, todavia a nota zero será atribuída;
- Evite discussões inócuas com o professor em tentar postergar a data de entrega do referido trabalho.