

Instruções

- (1) Realiza-se o Trabalho Prático 3 (TP2) **individualmente**.
- (2) O TP3 consiste em implementar e testar algoritmos de ordenação, e resolver problemas do URI Online Judge usando tais técnicas.
- (3) A parte proposta pelo professor vale 15 pontos. Os arquivos a serem submetidos devem estar dentro de um diretório de nome *matricula-nome*. Os devidos arquivos são os fontes *matricula-nome.cpp* e *matricula-nome.hpp* (caso necessário), um *makefile* com os devidos comandos de compilação, link-edição e execução, e um *relatorio.pdf* com o relato solicitado. Ao final, compacte o diretório no formato zip e submeta ao sistema ava.cefetmg.br até a data limite.
- (4) O exercício do URI vale 10 pontos. Será disponível 1 opção que deve ser feita APENAS no dia 9/6/2017 durante a aula. Não será permitido qualquer consulta em sites da Internet. É também proibido o acesso ao fórum de discussões. **Somente será permitido submeter a solução.** QUALQUER ACESSO INDEVIDO A OUTRA LOCALIDADE POR VIA QUALQUER DISPOSITIVO TERÁ A NOTA DO TRABALHO PRÁTICO 3 ANULADO!
- (5) A pontuação de cada problema será dado pela concordância do que foi solicitado e implementado, e pela taxa de acerto obtida no mesmo.
- (6) As soluções devem ser autênticas. Logo, não é permitido cópia parcial ou total de soluções entre alunos ou da Internet. Trabalhos copiados serão anulados¹.
- (7) É proibido usar funções de ordenação da linguagem e qualquer estrutura de dados ou funções da STL (salvo se o professor indicar).
- (8) No cabeçalho de cada código-fonte deve haver um conjunto de comentários com:
 - (a) O número de matrícula e o nome do(s) integrante(s) da equipe.
 - (b) Número e nome do problema URI solucionado pelo código-fonte.
 - (c) Paradigma de programação utilizado para resolver o problema.
 - (d) Descrição da solução desenvolvida.
 - (e) Ordem de complexidade do algoritmo desenvolvido.

Exemplo:

```
1  /* ALUNO: 201522126378 – Alain Lefevre
2  *
3  * PROBLEMA: 1055 – Soma Permutada Elegante
4  *
5  * PARADIGMA: Divisao e conquista
6  *
7  * DESCRICAO DA SOLUCAO: Ordena-se a metade do vetor com os maiores valores e a outra
8  * metade com os menores valores, depois obtém-se a melhor soma permutada.
9  *
10 * ORDEM DE COMPLEXIDADE: O(n log n)
11 */
```

Perde-se 1 ponto do total acumulado por errar a ordem de complexidade e por omitir, pelo menos parcialmente, alguma informação do cabeçalho.

- (9) Não se aceitará trabalhos entregues após a data especificada.

¹Será aplicado o sistema antiplágio MOSS (Measure Of Software Similarity), desenvolvido por Alex Aiken (Stanford).

Problema Proposto pelo Professor

(1) Realize as seguintes tarefas:

- (A) (5 ponto) Implemente em C++ as seguintes funções de ordenação total e crescente de vetores inteiros: Selection, Insertion, ShellSort, MergeSort, QuickSort1 (pivô centro da partição), QuickSort2 (pivô mediana entre três elementos da partição), QuickInsertion Sort (pivô mediana entre três elementos da partição, se a dimensão da partição for menor ou igual a 50 execute o inserção sobre a mesma).
- (B) (1 ponto) Implemente uma função que receba como parâmetros, a dimensão e uma referência de um vetor de inteiros vazio para povoá-lo com números aleatórios entre 0 e 20.000. Para a geração de número aleatórios será permitido apenas o uso das bibliotecas `random` através das classes `mt19937` e `uniform_int_distribution`. Use a função `chrono::high_resolution_clock::now().time_since_epoch().count()` da biblioteca `chrono` para inicializar a semente do gerador de números aleatórios.
- (C) (2 ponto) Implemente uma função que receba como parâmetros, a dimensão e o vetor preenchido com inteiros, e que realize um experimento computacional envolvendo todos os algoritmos de ordenação. Cada algoritmo deve ordenar o vetor passado como parâmetro e ter o devido tempo de execução, em milissegundos, armazenado (tome cuidado com o tipo para armazenar o tempo).
- (D) (2 ponto) Implemente e realize um experimento computacional para ordenar vetores com dimensões 10, 20, 50, 100, 200, 500, 1000, 5000. Execute 30 experimentos para cada dimensão/algoritmo com vetores distintos. Armazene os tempos em formato de tabela em um arquivo csv, de nome `saida.csv`, com a seguinte formatação:

```
Algoritmo,Dimensao,Tempo
Selection,10,123
Selection,10,119
Selection,10,203
...
Selection,5000,203987645445533
Insertion,10,89
...
```

- (E) (2 pontos) Externamente, usando um software de análise de dados como o R-project, gere as seguintes saídas:
 - a) Gráfico da mediana do tempo pela dimensão (um gráfico apresentando a evolução de todos os algoritmos).
 - b) Tabela cujas linhas são os algoritmos e as colunas são estatísticas quantil 0.1, 0.25, 0.5, 0.75 e 0.9, respectivamente.
- (F) (3 pontos) Construa um relatório, de no máximo 5 páginas, com as seções Introdução, Definição do Problema, Planejamento e Execução do Experimento, Resultados e Análises, e Considerações Finais. Será considerado aqui a capacidade de organizar as ideias, explicar a problemática, o desenvolvimento do trabalho e a capacidade de analisar os resultados. Cada erro de português decrementa 0,5 ponto.

Obrigatório: Execute o experimento em um único computador dedicado, desconectado da Internet e sem outros processos abertos. Não utilize-o enquanto o experimento é realizado, pois adulterará o resultado. Se possível, desligue o anti-vírus e outros softwares que podem executar de forma automática.