

Segundo Trabalho Prático

Ciência da Computação

1º Semestre de 2019

Este trabalho tem como objetivo permitir aos alunos a familiarização com os códigos dos algoritmos de ordenação vistos em sala (Ordenação por Seleção, por Inserção, *Shellsort*, *Quicksort*, *Heapsort* e *Mergesort*), através do trabalho sobre o código dos mesmos para avaliar o tempo de execução em diversas situações. Assim, você deverá implementar os algoritmos de ordenação citados e, considerando os seguintes tamanhos de vetores:

1. 20 elementos;
2. 500 elementos;
3. 5000 elementos;
4. 10000 elementos;
5. 200000 elementos;

as seguintes possibilidades de tamanhos de registros:

- (a) Registro pequeno (apenas a chave, inteira);
- (b) Registro grande no mínimo 50 campos com strings de 50 caracteres além da chave (inteira);

e as seguintes possibilidades de ordenação inicial de cada vetor de entrada:

1. Ordem aleatória (utilizar alguma função para geração randômica de chaves);
2. Ordenado crescentemente;
3. Ordenado decrescentemente.

Deverá analisar:

- (a) Tempo de execução;
- (b) Número de comparações;
- (c) Número de movimentações (ou seja, atribuições envolvendo elementos do vetor de entrada).

Repetir a execução, no mínimo, 10 vezes para cada condição/algoritmo. O resultado final deverá ser a média das execuções. Você deverá interpretar esses resultados (análise crítica), comparando-os com resultados teóricos (por exemplo, aqueles constantes nos slides). Abaixo, em “o que deve ser entregue”, há mais informações sobre o que deve ser feito.

Observações sobre a entrega:

- (a) O trabalho pode ser feito em grupos de, **no máximo**, três pessoas.
- (b) A parte de implementação do trabalho deverá ser entregue em um único arquivo compactado, com o nome dos integrantes (por exemplo, Fulano_de_Tal_Beltrano_de_Qual.zip). Indique, em um arquivo “leiam.txt”, em qual Sistema Operacional você desenvolveu o trabalho, assim como instruções para compilação. Caso tenha desenvolvido no DevC++ (Windows), utilizando projeto, envie o arquivo .dev também.
- (c) Nesse zip não deve haver arquivos executáveis.
- (d) Incluir pdf da parte escrita no zip.
- (e) A entrega dos arquivos deverá ser feita via moodle e a fórmula para desconto por atraso na entrega é $\frac{2^{d-1}}{0,32}\%$, onde d é o atraso em dias. Note que após 6 dias, o trabalho não pode ser mais entregue. Ao final da descrição do trabalho, há outras informações disponíveis sobre a entrega.

(f) A avaliação será individual: marcaremos uma data para a entrevista.

(g) **Data de entrega: 23/05/2019**

(h) Valor: 15 pontos

O que deve ser entregue:

- Documentação do trabalho (impressa). Em entre outras coisas, a documentação deve conter:
 1. Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
 2. Implementação: descrição sobre a implementação do programa.
 3. Resultados e Discussões: dados obtidos das execuções dos algoritmos de ordenação sobre cada um dos conjuntos de dados (incluindo gráficos e tabelas); análise crítica dos dados obtidos em comparação com o esperado.
 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 sites da Internet, se for o caso.
- Além disso, neste trabalho deve ser enviado ao professor o arquivo fonte. A entrega deverá ser feita via moodle, seguindo as diretrizes informadas no início da descrição deste trabalho.

Comentários Gerais:

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
- Clareza, identificação e comentários no programa também vão valer pontos.
- Avaliarei com maior atenção ainda a parte escrita - incluindo erros de português -. Atenção especial será dada à análise crítica.
- Trabalhos copiados serão penalizados com a nota zero.