

Trabalho Prático 2

Laboratório de Algoritmos e Estruturas de Dados I
Prof^a. Natália Batista

Nomes: _____ Total: 15 pontos

Instruções

O trabalho prático deverá ser realizado em duplas e poderá ser entregue até o dia 29/11/2019, por *email*. O prazo de entrega deste trabalho NÃO será prorrogado.

Este trabalho deverá conter uma parte escrita conforme descrita adiante e seguindo as diretrizes do Anexo I. A parte prática deverá ser implementada em linguagem C ou C++ no ambiente Linux. Lembre-se de incluir as instruções de compilação dos programas e eventuais arquivos auxiliares, como “makefiles”.

Para a entrega dos arquivos (documentação e códigos), siga as instruções contidas no Anexo II.

O enunciado deste trabalho foi adaptado de um dos trabalhos da disciplina de Projeto e Análise de Algoritmos do Prof. David Menotti Gomes (DECOM/UFOP).

Implementação e Análise de Algoritmos de Ordenação por Comparações de Chaves

1. Descrição

Neste trabalho, você deverá implementar algoritmos não-eficientes (BubbleSort, Inserção e Seleção) e eficientes (QuickSort, HeapSort e MergeSort) para ordenar sequências de elementos.

Você deverá utilizar implementações dos algoritmos citados onde a sequência de dados a ser ordenada pode estar representada por arranjos (vetores) ou apontadores. Lembre-se que as implementações dos algoritmos de ordenação estão disponíveis nos slides de aula da disciplina teórica (https://sites.google.com/site/nataliacefetmg/home/aed1_20192) e deverão ser utilizadas.

Cada elemento da sequência de dados a ser ordenada deve conter um campo chave e pelo menos um campo adicional à sua escolha. A especificação e desenvolvimento dos tipos abstratos de dados a serem utilizados no trabalho deverão ser baseados nas implementações do livro texto da disciplina (<http://www2.dcc.ufmg.br/livros/algoritmos/implementacoes.php>).

2. Entrada e saída do programa

O programa principal deverá receber opções da linha de comando (parâmetros *int argc* e *char **argv* da função *main*) e executar conforme as opções escolhidas, que são:

- **-a:** algoritmo a ser executado. O algoritmo pode ser:

Bubblesort
Insercao
Selecao
Quicksort
Heapsort

Mergesort

- **-n:** número de elementos do vetor.
- **-t:** tipo de entrada. O programa deverá gerar um vetor de entrada de um dos tipos a seguir:

1 - em ordem crescente (ordenado)

2 - em ordem decrescente

3 - aleatório

4 - quase ordenado

- **-v:** imprimir vetor original sim (S) ou não (N). Imprime somente as chaves dos registros.
- **-r:** imprimir vetor ordenado sim (S) ou não (N). Imprime somente as chaves dos registros.

Exemplo: para realizar a execução do Quicksort com um vetor com 100 elementos em ordem crescente e também imprimir na tela o vetor original e o vetor ordenado, o comando deverá ser

```
./a.out -a Quicksort -n 100 -t 1 -v S -r S
```

A saída do programa será a impressão na tela do tempo de execução (tempo de usuário) no formato:

Tempo de usuário: X segundos e XX microssegundos

3. Como gerar os vetores/arranjos para ordenação

Considere sequências de dados com a quantidade de elementos variando, i.e., 10, 100, 1000, 10000, 100000, 1000000, etc. Considere também sequências sem valores de chaves repetidos. Ainda, considere que as chaves dos elementos das sequências correspondem a valores inteiros e as sequências podem estar **ordenadas**, **inversamente ordenadas**, **quase ordenadas (20% dos elementos fora de ordem)** e **aleatórias** em relação aos valores das chaves de seus elementos.

4. O que analisar

A análise deverá ser feita sobre o tempo de execução dos algoritmos e sua relação com a complexidade de tempo em relação ao número de comparações. Para obter o tempo de execução, deverão ser feitas no mínimo 3 execuções distintas e obtida a média dos tempos de usuário. Organize os dados coletados em tabelas sucintas e também construa gráficos relevantes a partir dos dados (não colocar tempos relativos nas tabelas e gráficos). Então, disserte sobre os dados nas tabelas e gráficos. Grande parte da avaliação será dedicada à análise dos resultados, ou seja, sobre o que você dissertar.

Para comparação, relacione somente os algoritmos não-eficientes e eficientes entre si. Por exemplo, não relacione, analise ou compare, o algoritmo BubbleSort com o QuickSort.

5. O que deverá ser entregue

- Código fonte do programa em C ou C++ (bem indentado e comentado).

- Documentação do trabalho (um relatório com no máximo 6 páginas com fonte tamanho 12, incluindo todas as partes como, por exemplo, capa) no formato PDF.

A documentação deverá conter obrigatoriamente as seguintes seções:

1. Introdução: descrição do problema a ser resolvido e análise de complexidade (estudo da complexidade de tempo das funções implementadas no pior caso utilizando a notação O). Deve constar a prova do resultado obtido ou a referência da prova. (Máximo 1 página)

2. Implementação: visão geral sobre o funcionamento do programa. Deverão ser detalhadas as estruturas de dados utilizadas (de preferência com diagramas 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 NÃO deverão ser inseridos na documentação. (Máximo 1 página)

3. Experimentos e análise de resultados: explicação de como os vetores foram gerados e a metodologia usada nos experimentos. Os testes executados devem ser apresentados, analisados e discutidos (lembre-se de incluir as configurações da máquina utilizada). (Máximo 2 páginas)

4. Conclusão: comentários gerais sobre o trabalho e análise das principais dificuldades e limitações encontradas em sua implementação e nos experimentos. (Máximo 1/2 página)

Referências: toda bibliografia utilizada para o desenvolvimento do trabalho, incluindo páginas da Internet, se for o caso. Uma referência bibliográfica deve ser citada no texto quando da sua utilização. (Máximo 1/2 página)

Formato final: mandatoriamente em PDF.

6. Como deverá ser feita a entrega

A entrega deverá ser feita por e-mail, contendo o código fonte, arquivos diversos e a documentação. As instruções de entrega estão no Anexo II.

7. Comentários Gerais

- O trabalho poderá ser realizado em **duplas**.
- O trabalho será realizado durante as aulas práticas dos dias 08/11, 22/11 e 29/11.
- Trabalhos entregues em atraso não serão aceitos.
- O programa deverá permitir testes com quaisquer tamanhos de vetores, sendo que esses deverão ser gerados a cada execução (o tempo para gerar os vetores pode ser incluído na medição do tempo total).
- O programa deverá ser bem organizado, indentado e conter comentários explicativos relevantes.
- Os testes do programa entregue serão realizados em ambiente **Linux**. O programa deverá compilar sem erros ou avisos (*warnings*) com o compilador *gcc*. Programas que não compilarem ou que apresentarem erros de execução (falhas de segmentação, etc) não serão corrigidos.

- Não é permitida no texto da parte escrita a cópia de trechos de livros, sites e outros materiais. Caso seja necessário fazer referência a trabalhos de outros autores, é obrigatório fazer as citações corretamente. Se houver indícios de plágio na parte escrita ou no código fonte do programa, a nota final do trabalho será zero e serão aplicadas as penalidades previstas no Regime Disciplinar Discente.
- A parte escrita do trabalho deverá ter no **máximo 6 páginas com fonte tamanho 12**. A utilização do *template* Latex fornecido é opcional (mesmo usando o *template*, o limite de páginas total continua o mesmo). Resumo, sumário e anexos não são necessários. O trabalho deverá conter obrigatoriamente as seções citadas anteriormente:
 - Capa (máximo 1 página)
 - 1. Introdução (máximo 1 página)
 - 2. Implementação (máximo 1 página)
 - 3. Experimentos e análise de resultados (máximo 2 páginas)
 - 4. Conclusões (máximo 1/2 página)
 - Referências (máximo 1/2 página)
- **Trabalhos com mais de 6 páginas não serão corrigidos - a nota será zero (0)**. O trabalho deverá conter obrigatoriamente as seções descritas anteriormente e com o mesmo título das mesmas. Não envie a parte escrita desmembrada em mais de um arquivo, pois só será corrigido 1 (um) PDF de cada dupla.
- Não serão aceitos '*prints*' de caixas de *email* como prova de envio do trabalho. Somente emails confirmados pela professora serão aceitos. Verifique os anexos, a confirmação é somente do recebimento do email e não do seu conteúdo.

Bom trabalho!