

Data de entrega: 24/11/2011

Leia atentamente todas as instruções abaixo:

- 1) Um (1,0) ponto será descontado na nota para cada dia de atraso na entrega.
- 2) O trabalho deve ser desenvolvido em grupos de 2 pessoas.
- 3) Enviar o trabalho para o email do docente:
lantiq@icmc.usp.br ou lantiq@gmail.com (assunto da mensagem: "[ICC-II] Entrega TP2")
Uma mensagem de confirmação de entrega será enviada pelo professor ao(s) aluno(s).
- 4) Não anexar no email arquivos executáveis. Inclusive, verifique todas as sub-pastas de arquivos do tipo zip, rar, etc.
- 5) Não deixe para entregar o trabalho na última hora. A responsabilidade de entrega é do aluno, e não do servidor de email. A entrega por email facilita o processo, mas não garante 100% de confiabilidade. Se, por algum motivo, o servidor de emails estiver com problemas e você não receber a mensagem de confirmação de entrega, grave o trabalho em um CD e entregue-o na sala do docente (3-160). Nesse caso, identifique o CD com os nomes dos alunos, o conteúdo e a data de entrega. Deixe o CD embaixo da porta, caso o docente não se encontre na sala. Ou seja, faça com que o trabalho chegue às mãos do professor.

Especificação do Trabalho

Crie uma nova biblioteca para a linguagem C com todos os métodos de ordenação vistos em aula. Para tanto, crie um arquivo `sorting.c`, o qual deverá conter as implementações dos seguintes métodos:

- a. **Insertion sort** (inserção simples)
- b. **Selection sort** (seleção direta, versão melhorada)
- c. **Bubble sort** (versão melhorada)
- d. **Heap sort**
- e. **Merge sort**
- f. **Quick sort**
- g. **Shell sort**

Para este projeto, utilize SOMENTE as versões desses métodos dadas em aula, NÃO desenvolva ou colete implementações de outras fontes.

Crie um arquivo header (`sorting.h`) contendo os protótipos das funções de ordenação definidas em `sorting.c`.

A seguir, crie um programa principal que irá aplicar essa biblioteca (use `#include "sorting.h"`) da seguinte maneira:

1. Peça ao usuário que entre com um número n inteiro estritamente positivo.
2. Aloque dinamicamente um vetor v de n elementos que contenha números INTEIROS definidos aleatoriamente. *Pesquise como gerar corretamente números aleatórios em C.*
3. Aplique cada um dos métodos de ordenação no vetor v . Tome cuidado para que o arranjo original do vetor v seja mantido, já que a aplicação de um método de ordenação altera o vetor passado como parâmetro. Imprima na tela o tempo de execução de cada algoritmo (esta é uma medida aproximada do caso médio). *Pesquise como ler automaticamente o relógio do computador por meio da linguagem C.*
4. Aplique cada um dos métodos de ordenação no vetor v que acabou de ser ordenado (melhor caso). Imprima na tela o tempo de execução de cada algoritmo nesse caso.
5. Crie uma função que inverta os elementos de um vetor. Inverta o vetor v já ordenado, de modo que ele fique inversamente ordenado. A seguir, aplique cada um dos métodos de ordenação no vetor v inversamente ordenado (pior caso). Imprima na tela o tempo de execução de cada algoritmo nesse caso.

Tenha especial atenção em exibir os resultados na tela de maneira bem organizada. *Execute os passos anteriores para diversos tamanhos de vetor. Tome n iniciando em n_0 e aumente-o em incrementos iguais a Δn . Você deverá realizar esses incrementos 19 vezes, ou seja:*

$$n = n_0, n_0 + \Delta n, n_0 + 2\Delta n, n_0 + 3\Delta n, \dots, n_0 + 19\Delta n$$

Escolha n_0 e Δn de modo que seja possível analisar as diferenças no tempo de execução dos algoritmos de ordenação considerados. Sua documentação externa deverá conter três tabelas do tipo:

	$n = n_0$	$n = n_0 + \Delta n$	$n = n_0 + 2\Delta n$	$n = n_0 + 3\Delta n$		$n = n_0 + 19\Delta n$
Insertion sort					...	
Selection sort						
Bubble sort						
Heap sort						
Merge sort						
Quick sort						
Shell sort						

a qual exibirá os tempos de execução dos algoritmos de ordenação, sendo uma tabela para cada caso (ou seja, uma para o caso médio, outra para o melhor caso e outra para o pior caso).

Você deverá entregar:

- 1) O código-fonte, com comentários explicando as principais funcionalidades do seu programa. Também preste atenção à correta indentação de seu código.
- 2) A documentação externa do seu trabalho, incluindo:
 - a. Passo-a-passo para compilação do projeto (incluindo versão do compilador).
 - b. Especificações da máquina onde o projeto foi executado (processador, quantidade de memória RAM e sistema operacional).
 - c. As três tabelas acima especificadas.
 - d. Interpretação dos resultados obtidos: comente os tempos de execução observados e os relacione às respectivas análises assintóticas vistas em aula.