

## ***Algoritmos de Ordenação***

### **1- Objetivo**

O **objetivo** deste trabalho é implementar alguns dos algoritmos de ordenação estudados, comparando-os entre si.

### **2 - Implemetação (Parte I)**

Inicialmente, o seu trabalho é criar um programa para geração automática de números em **C**. Este programa, doravante denominado *gera*, deverá se comportar da seguinte forma:

- quando digitarmos  
. \gera -a n  
o programa deverá gerar n números em ordem aleatória.
- quando digitarmos  
. \gera -c n  
o programa deverá gerar n números em ordem crescente.
- quando digitarmos  
. \gera -d n  
o programa deverá gerar n números em ordem decrescente.

Os números gerados deverão ser apresentados um por linha (sem espaço depois do número!). Segue um exemplo:

. \gera -c 5

1  
2  
3  
4  
5

Os dados obtidos com o algoritmo *gera* serão utilizados para testar os algoritmos de ordenação da seção seguinte.

### **3 - Implemetação (Parte II)**

O seu trabalho agora é implementar **em C** os seguintes algoritmos de ordenação: *bubblesort*, *shakesort*, *insertionsort*, *shellsort*, *selectionsort*, *ranksort* (sem comparação de chaves), *quicksort*, *mergesort*, *heapsort*, *radixsort* (sem comparação de chaves e

*utilizando o bucketsort) e radixsort binário (sem comparação de chaves e utilizando o bucketsort).*

### 3.1 - Quicksort

O algoritmo Quicksort utiliza um método de particionamento que, escolhido um elemento pivô, gera uma partição de elementos maiores ou iguais ao pivô e outra partição cujos elementos são menores ou iguais ao pivô.

Consideraremos, neste trabalho, as seguintes formas de escolher o pivô:

- O primeiro elemento do bloco;
- O elemento central do bloco;
- Um elemento aleatório;
- Mediana de 3 elementos, onde os elementos escolhidos para tal média serão o primeiro, o central e o último.

## 4 - Formatação de Entrada e Saída

O trabalho será testado da seguinte maneira:

**`./trab4 algoritmo n < entrada.txt`**

onde:

- *entrada.txt* será o arquivo com o resultado obtido com o algoritmo *gera* para *n* elementos;
- *n* é o número de elementos a serem ordenados
- *algoritmo* é o algoritmo que será utilizado na ordenação. As opções são: *bubble*, *shake*, *insertion*, *shell*, *selection*, *rank*, *quickprimeiro*, *quickcentral*, *quickrandom*, *quickmediana3*, *merge*, *heap*, *radix* e *radixbin*.

Você deverá ter como **saída** os números da entrada ordenados, sendo **cada número em uma linha** (sem espaço depois do número!).

Um exemplo da utilização do trabalho escolhendo-se o algoritmo Bubblesort para a ordenação de 10 números:

**`./trab4 bubble 10 < entrada.txt`**

Agora um exemplo escolhendo-se o algoritmo Quicksort não recursivo com escolha do pivô utilizando-se o método da mediana de 3 para a ordenação de 10 números:

**`./trab4 quickmediana3 10 < entrada.txt`**

## 5 - Documentação

**Como resultado deste trabalho deverá ser produzido um artigo técnico em LaTeX utilizando o padrão de artigos, e este deverá ser entregue também impresso.**

Na documentação deverão ser apresentadas comparações entre os algoritmos (com gráficos!), os tempos alcançados e os tipos das entradas propostas: aleatória,

crescente e decrescente. **Uma detalhada discussão sobre os resultados também deverá ser apresentada.**

## **6 - Avaliação**

- A nota do trabalho terá um total de 10 pontos.
- Serão contemplados com nota zero os trabalhos que se enquadrarem em uma ou mais situações abaixo:
  1. Plágio;
  2. Programa não compila;
  3. Não está de acordo com as especificações.

## **7 - Considerações Importantes**

- Modularize o seu código adequadamente. Crie arquivos .c e .h para cada módulo do seu sistema. Em especial, crie arquivos exclusivos para manipular as estruturas de dados dos tipos abstratos de dados que você estiver representando.

- Entrega

Este trabalho deve ser feito **INDIVIDUALMENTE** e entregue até o dia **02 de dezembro de 2015**, às **17:00hs**, em sala.

Ele deve ser enviado para:

[mberger@inf.ufes.br](mailto:mberger@inf.ufes.br).

O assunto da mensagem deve ser:

ed2:trab4:<nome1>

O nome do arquivo com o trabalho enviado em anexo deve ser trab4.tar.gz. Ele será descompactado da seguinte forma:

```
tar -xvzf trab4.tar.gz
```

e deverá gerar um arquivo chamado Makefile, com as regras de compilação do programa, os arquivos com os códigos dos programas e o arquivo com a documentação em LaTeX (que deverá ser chamado de artigo.tex).

Ao digitar:

```
make all
```

deve ser gerado os executáveis gera e trab4 e a documentação em PDF (trab4.pdf) a partir dos fontes em LaTeX

O recebimento dos trabalhos é automatizado. Siga as instruções à risca pois algum erro na submissão pode inviabilizar a entrega do seu trabalho.

**8 - Observação Importante**

**Mais detalhes serão discutidos em sala de aula. Considerações feitas em sala terão mais relevância do que as contidas nesta especificação.**