

Trabalho T1

Este trabalho tem como objetivo a prática de operações de ordenação simples, eficiente e lineares sobre estruturas de dados complexas.

I Instruções

O trabalho deve conter, para cada questão, um arquivo com o código exigido, comentários explicando a lógica de implementação e uma demonstração de funcionamento do código (função *main*).

As ordenações a serem implementadas devem ser realizadas sobre uma lista encadeada simples e o resultado ordenado deve ser armazenado numa fila. Os códigos para construir/desconstruir a lista encadeada simples e a fila, percorrê-las e imprimir seus elementos, devem ser desenvolvidos como parte do trabalho.

O estudante deve declarar um vetor inicial de inteiros para demonstrar o caso de uso e construir a lista encadeada simples. A ordenação somente pode ser iniciada após a construção da lista. Para realizar a ordenação após a construção da lista, fica a critério do estudante desconstruir a lista e colocar os nodos em um vetor ou se ordenará diretamente na lista encadeada simples. Após ordenados, os elementos devem ser armazenados na fila. *Se utilizada a opção pelo vetor, a ordenação deve ser feita movimentando os nodos e não os valores das chaves.*

Utilize o struct abaixo para criar os nodos necessários:

```
struct tlist {  
  
    int valor;  
  
    struct tlist* next;  
  
};  
  
struct tqueue {  
  
    int valor;  
  
    struct tqueue* next;  
  
};
```

II Entrega

A entrega da atividade será on-line através do SIGAA. O estudante deve enviar um único arquivo ZIP contendo as suas implementações e demais itens solicitados. O arquivo ZIP deve ser nomeado com a matrícula do estudante.

Data limite para entrega: 13/10/2024, 23h59m.

III Avaliação

A avaliação será realizada de acordo com a entrega do estudante observados os seguintes critérios:

1. código compilando e funcionando sem alterações na lógica;
2. nível de comentários explicando a lógica utilizada;
3. presença da demonstração do código;
4. construção da lista encadeada simples;
5. construção da fila;
6. utilização dos laços de repetição da forma correta (`for`, `while`, `do-while`, etc).

Em caso de plágio (incluindo utilização de ferramentas de geração de código), a entrega será desconsiderada e os estudantes envolvidos receberão nota 0. Procedimentos administrativos poderão ser iniciados após consulta à coordenação de curso. Os professores do CCR se reservam ao direito de solicitar que estudantes apresentem o código produzido para o trabalho após a entrega.

IV Atividades

01 – Implemente uma ordenação ineficiente à sua escolha.

02 – Implemente a ordenação eficiente do algoritmo *quick sort*.

03 – Implemente a ordenação linear *radix sort*.