



INSTITUTO FED. DE EDUCAÇÃO, CIÊNC. E TEC. DE PERNAMBUCO
CURSO: TEC. EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
DISCIPLINA: ALGORITMOS E ESTRUTURAS DE DADOS
PROFESSOR: RAMIDE DANTAS
ASSUNTO: LISTAS E BUSCA BINÁRIA.

Prática 04

Parte 1: Lista Simples

Problema 1: Implemente a função `remover()` da `Lista` no código da atividade (zip).

Teste a remoção com os casos de teste existentes. Adicione mais se necessário.

Desafio (Opcional): Ajustar o código da lista para resolver o problema [BC1740](#):

Problema 2: No `main.cpp`, crie a `struct` `compras`, contendo nome (`string`), quantidade (`int`), preço (`float`). Use a `struct` `tarefa` como referência.

Sobrecarregue (implemente) o operador `==` da `struct` `compras`, que compara um item do tipo `compras` baseado apenas no nome do item. Esse operador é usado na busca sequencial.

Na função `main()`, crie uma instância da lista de compras. Adicione itens à lista, faça inserções, e varra a lista, exibindo seu conteúdo (assim como fizemos com a lista de tarefas.)

Parte 2: Lista Ordenada

Problema 1: Implemente a função `buscar()` da `ListaOrdenada` usando a busca binária de forma **recursiva** ou **iterativa**.

Faça testes com a lista de inteiros e de tarefas e verifique se o resultado é o esperado.

Mude a lista de compras para a lista ordenada e faça novos testes.

Será necessário sobrecarregar os operadores `<` e `>` da `struct` `compras` de forma a comparar os itens usando apenas o nome. Veja a `struct` `tarefa` como referência.

Desafio (Opcional): mude o critério de ordenação para ser o preço total (preço * quantidade).

Problema 2: Resolva o problema [LC703](#) no LeetCode usando `ListaOrdenada`.

Nesse problema, a classe `KthLargest` mantém as pontuações de uma universidade fictícia, e deve retornar o K-ésimo **maior** valor (ex., o 1º ou 3º) após a inserção de uma nova pontuação via função `add()`. A instância da classe `KthLargest` é inicializada com o valor `k` e um `vector<int>` de valores (pontuações) iniciais. Use a `ListaOrdenada` para resolver esse problema (ver código abaixo). Observações:

- Faça uso de instância(s) da `ListaOrdenada`; não copie seu código para dentro de `KthLargest`. Não é preciso fazer alterações na `ListaOrdenada` se as partes anteriores estiverem corretas.
- Lembre que a `ListaOrdenada` está em ordem crescente, do menor para o maior.
- Para submeter o código no LeetCode (**opcional**), é preciso copiar e colar o conteúdo de `lista.h` junto com o código da classe `KthLargest`, e não incluir o código da função `main()`.

```
#include <iostream>
#include "lista.h"

class KthLargest {
    // TODO
public:
    KthLargest(int k, vector<int>& nums) : k(k), scores(10000) {
        // TODO
    }

    int add(int val) {
        // TODO
    }
};

int main() {
    vector<int> nums = {4, 5, 8, 2};
    vector<int> vals = {3, 5, 10, 9, 4};
    KthLargest kthLargest (3, nums);

    for (int v : vals) {
        cout << "Adicionando " << v << ", 3o maior eh " << kthLargest.add(v)
    << endl;
    }
}
```