

Algoritmos e Estrutura de Dados PCO001

Resolução de: **Exercícios - Listas**

Carlos Augusto Ribeiro

2024-01

Instruções para execução da atividade:

- (a) Para todos os exercícios propostos utilizar as implementações baseadas em **ponteiros**.
- (b) Caso seja necessário, altere o tipo do dado a ser utilizado (e.g., **char** para **int**).
- (c) A função implementada em uma tarefa pode ser utilizada na solução de outra (e.g., utilizar TamanhoLista para calcular a média dos valores de uma lista).

Tarefa 1 - Implemente uma função que verifica se um dado elemento inteiro pertence ou não a lista. Protótipo: bool MembroLista (NoPtr, int).

Solução

```
bool MembroLista(NoPtr no, int target) {  
    if (no == nullptr) return false;  
    if (no->Info == target) return true;  
    return MembroLista(no->Lig, target);  
}
```

Tarefa 2 - Implemente uma função para retornar o último elemento de uma lista. Protótipo: NoPtr UltimoDaLista (NoPtr).

Solução

```
NoPtr UltimoDaLista(NoPtr no) {  
    if (no->Lig == nullptr) return no;  
    return UltimoDaLista(no->Lig);  
}
```

Tarefa 3 - Implemente uma função para retornar o número de elementos em uma lista. Protótipo: `int TamanhoLista (NoPtr)`.

Solução

```
int TamanhoLista(NoPtr no) {  
    if (no == nullptr) return 0;  
    return TamanhoLista(no->Lig) + 1;  
}
```

Tarefa 4 - Implemente uma função para retornar a média dos elementos de uma lista. Protótipo: `float MediaLista (NoPtr)`.

Solução

```
float MediaLista(NoPtr no) {  
    NoPtr aux = no;  
    float soma = 0;  
    while (aux != nullptr) {  
        soma += aux->Info;  
        aux = aux->Lig;  
    }  
    return soma / TamanhoLista(no);  
}
```

Tarefa 5 - Implemente uma função que retorna se duas listas são iguais ou não. Para serem consideradas iguais as listas precisam ter os mesmos elementos e ter o mesmo tamanho. A Figura 1 apresenta um exemplo do funcionamento dessa função. Protótipo: `bool ListasIguais (NoPtr, NoPtr)`.

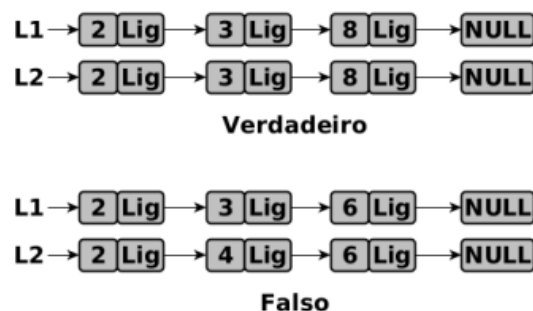


Figure 1: Exemplo do funcionamento da função `ListasIguais`.

Solução

```

bool ListasIguais(NoPtr l1, NoPtr l2) {
    if (l1 == nullptr && l2 != nullptr) return false;
    if (l2 == nullptr && l1 != nullptr) return false;
    if (l1 == nullptr && l2 == nullptr) return true;
    if (l1->Info != l2->Info) return false;
    return ListasIguais(l1->Lig, l2->Lig);
}

```

Tarefa 6 - Implemente uma função que unifica (merge) duas listas em uma nova lista. A Figura 2 apresenta um exemplo do funcionamento dessa função. Protótipo: NoPtr MergeListas (NoPtr, NoPtr).

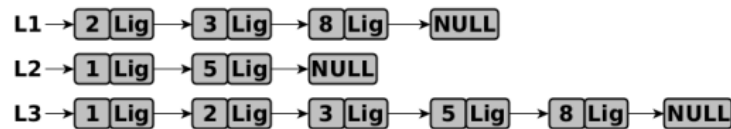


Figure 2: Exemplo do funcionamento da função MergeListas.

Solução

```

NoPtr MergeListas(NoPtr l1, NoPtr l2) {
    NoPtr novaLista = nullptr;
    NoPtr aux = l1;
    while (aux != nullptr) {
        if (!MembroLista(novaLista, aux->Info))
            InsereLista(novaLista, aux->Info);
        aux = aux->Lig;
    }
    aux = l2;
    while (aux != nullptr) {
        if (!MembroLista(novaLista, aux->Info))
            InsereLista(novaLista, aux->Info);
        aux = aux->Lig;
    }
    return novaLista;
}

```