Estrutura de Dados I

Lista (Encadeada ou em inglês, Linked List)

Matheus Gabriel

Outubro 2024

1 Definição

Costuma ser chamada apenas de lista, ela é:

1. Conjunto de dados do mesmo tipo
2. **Não** Usamos indices para acessar os seus elementos, usamos **nós**

2 Nós (nodes)

Para acessar elementos na lista encadeada usamos **nós**, eles são alocados dinamicamente, conforme a necessidade. E também eles não precisam estar localizados sequencialmente na memória.

3 Vantagens e Desvantagens

Vantagens:

1. Tamanho dinâmico
2. Não há deslocamento de memória na inserção de elementos
3. Remoção de elementos NÃO deixam "buracos" nos arrays

Desvantagens:

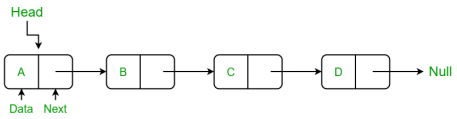
1. Nós não são sequenciais na memória
2. Custo extra de memória (*Overhead*)

4 Composição

Um nó da lista contém duas partes:

1. data → os dados armazenados no nó
2. *next → um ponteiro/referência para o próximo nó da lista (já que nós são armazenados sequencialmente na memória).

Visualmente eles podem serem representados como:



5 Implementação

Não é necessária a criação de uma classe Linked List, pode ter apenas o Nó *head*.

5.1 Apenas o nó

5.1.1 C

```c
struct Node {
    char data;
    struct Node* next;
};
```

Repare que next é um ponteiro para outro Node (pode ser qualquer Node, incluindo ele mesmo). O tipo char no data é só um exemplo.

5.1.2 Java

```java
public class Node {
    private char data;
    private Node next;

    // Constructor, getter, setter, etc...
}
```

Mais simples, porém a implementação é semelhante.

5.2 Usando a classe Linked List

```
public class LinkedList {
    // Primeiro no
    private Node head;

    // OPCIONAL
    // Ultimo no
    private Node tail;
    // Contador de nos
    private int count;
}
```

Na verdade, como nos códigos anteriores, a única parte necessária é o primeiro nó.