

Tipos abstratos de dados

Listas

Prof. Tiago Massoni
Prof. Fernando Buarque
Prof. Byron Leite

Engenharia da Computação

Poli - UPE

Noção de tipos abstratos de dados (TAD)

- TAD = Valores + Operações sobre estes valores
- TADs implicam em conceitos matemáticos (*i.e.* algebras) que são independentes das suas implementações

Noção de tipos abstratos de dados (TAD)

- A implementação do algoritmo em uma linguagem de programação específica exige a representação do TAD em termos dos tipos de dados e dos operadores suportados
 - A representação do modelo matemático por trás do TAD é realizada mediante uma estrutura de dados

TADs

- Exemplo: o conjunto dos inteiros (objetos) acompanhado das operações de adição, subtração e multiplicação
- Em Java
 - Classes: oferecem operações sobre um tipo
 - Encapsulamento: não sabemos como serão implementados

Listas

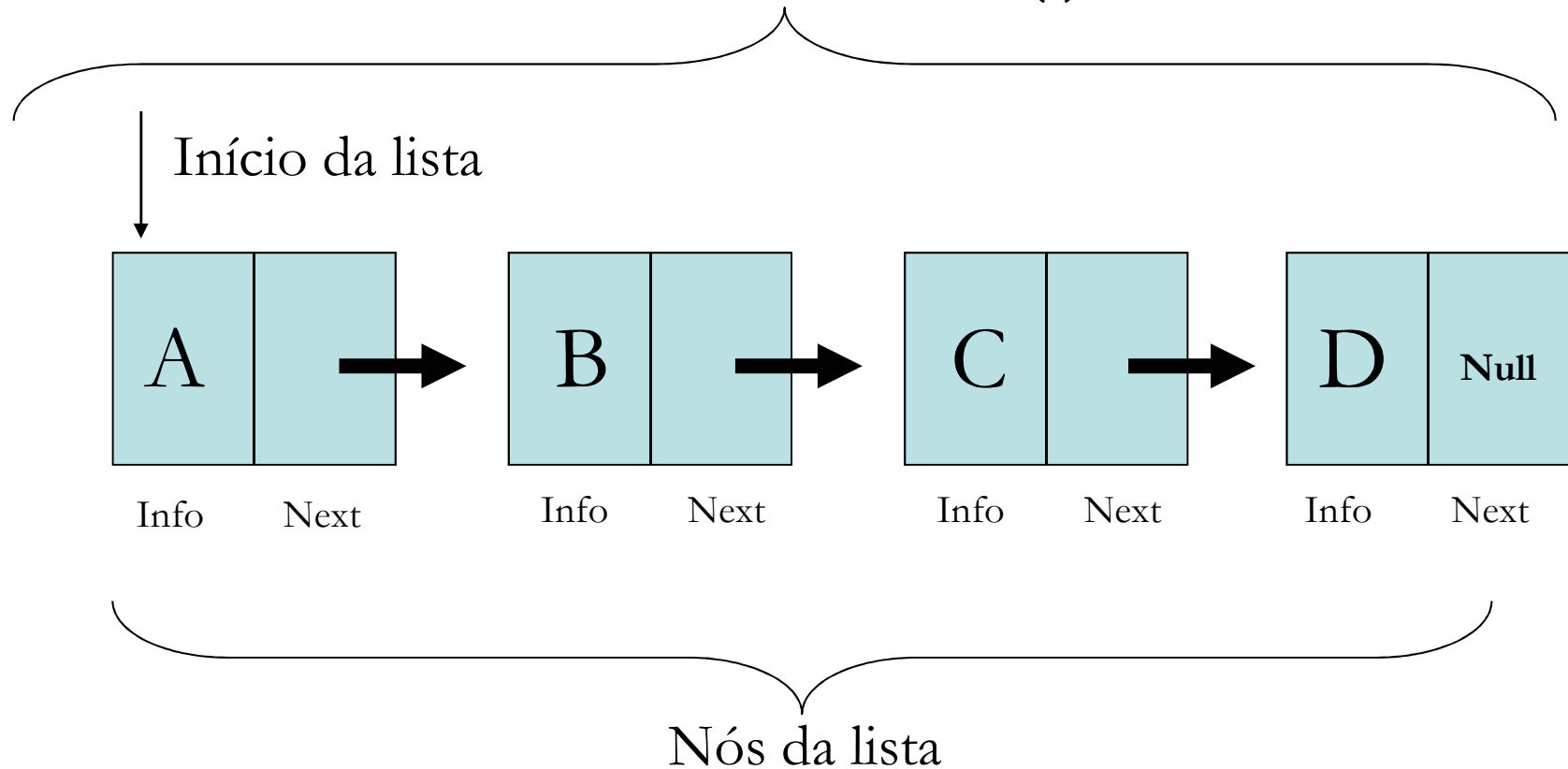
“Listas são conjuntos de dados (i.e. estruturas de dados) nos quais existe um encadeamento sequencial dos elementos. Novos elementos podem ser inseridos ou elementos pré-existentes podem ser removidos a depender da ordem especificada para o conjunto

O TAD "Lista"

- $A_0, A_1, A_2, \dots, A_{(n-1)}$ – lista geral
 - N: tamanho da lista ($n=0$, lista vazia)
- A_{i+1} sucede A_i , que sucede A_{i-1}
 - Posição = i
- Operações
 - imprimeLista
 - esvaziar
 - inserir, remover
 - procurarIEsimo

Lista - intuição

Lista linear - *List* (1)



Listas

- **Nó node p**
 - Informação - conteúdo útil armazenado **p.info**
 - Next - refer. para o nó seguinte **p.next**

Observação: a referência nula (**null**) indica o final da lista

- **Lista vazia**
list = null
- **Se $p.next \neq null$ então**
 $p.next.info$ é a porção informação
do nó posterior ao nó apontado por p

Listas com arrays

```
public class ListaArrays {  
    private Object[] array;  
    private int count=0;  
    public ListaArrays(){  
        this.array = new Object[MAX];  
    }  
    ...  
    public void printList(){..}  
    public Object find(Object x){..  
    public Object findKth(int pos){..  
    public void insert(Object x){..  
    public void remove(Object x){..  
}
```

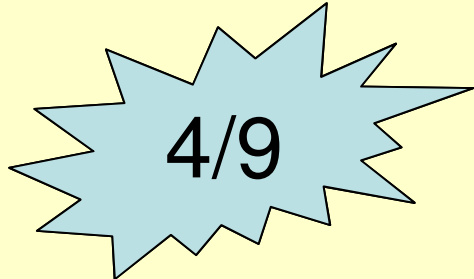
Listas com arrays

- findKth - tempo constante
- printList e find - sempre o mesmo tempo para o mesmo número de nós
- No entanto, inserir e remover são extremamente custosos (no caso da lista ordenada)
- Outras desvantagens
 - Tamanho não pode ser alterado durante execução
 - quantidade fixa de elementos
 - Ex.: impossível incluir um elemento a mais que MAX (sub-utilização)
 - Estrutura com menos elementos que MAX ocupam toda a memória pré-alocada

Exercício

1. Implemente a classe **BIGNUM** para números inteiros, contendo as operações abaixo.

```
public class BIGNUM {  
    private String digits;  
    private bool signal;  
    public BIGNUM(String n) {  
        // ...  
    }  
    ...  
    public void soma(BIGNUM o) {...}  
    public void subtrai(BIGNUM o) {...}  
    public void multiplica(BIGNUM o) {...}  
    public BIGNUM divide(BIGNUM o) {...}  
    public void potencia(int p) {...}  
    public String toString() {...}  
    public int compareTo(Object o) {...}  
}
```



4/9