Pilhas

Prof. Byron Leite

Prof. Tiago Massoni
Prof. Fernando Buarque

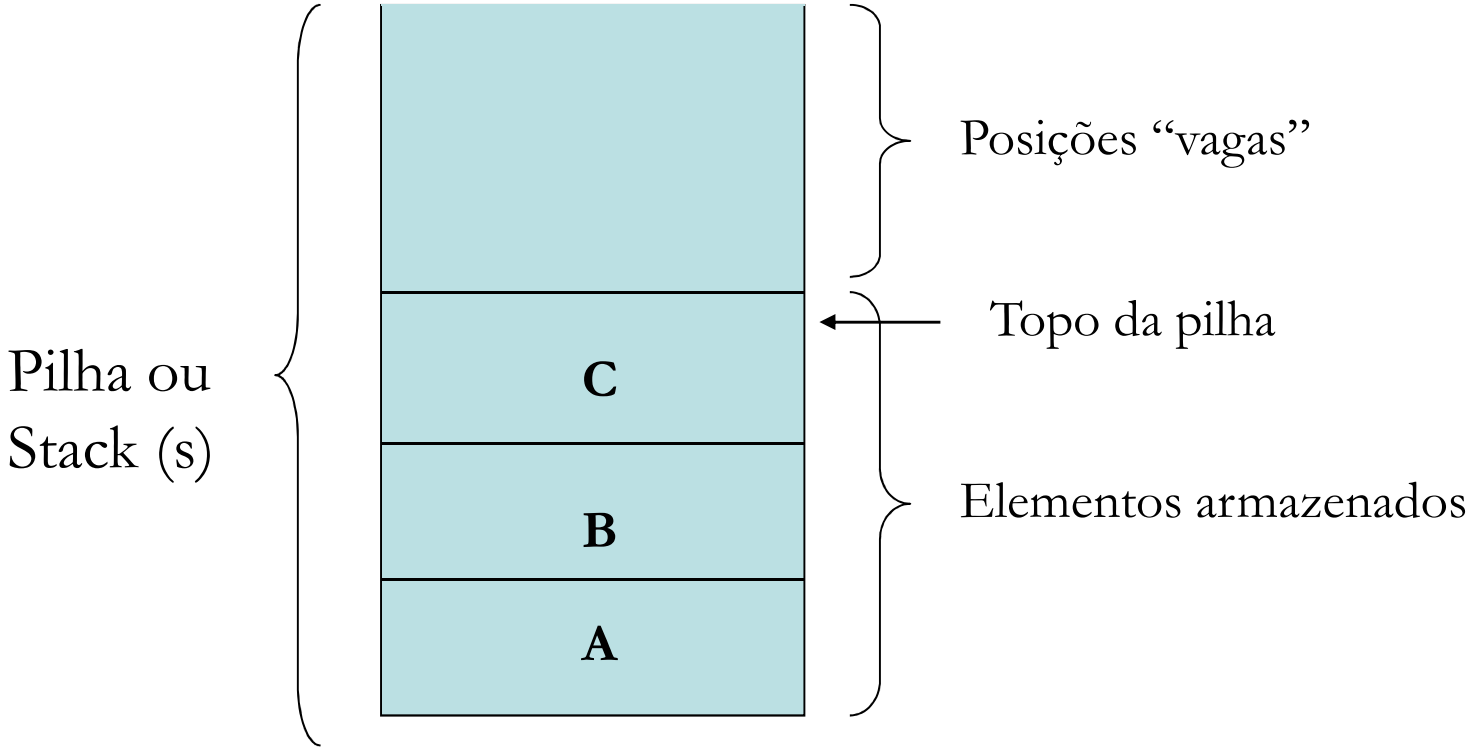
Engenharia da Computação

Poli - UPE

Definição

"**Pilhas** são conjuntos ordenados de dados (i.e. estruturas de dados) nos quais novos elementos podem ser inseridos ou elementos pré-existentes podem ser removidos sempre de uma extremidade, i.e. o topo da pilha"

Intuição


Pilha ou
Stack (s)
C
B
A
Posições "vagas"
Topo da pilha
Elementos armazenados

Características

- Pilhas se expandem ou se reduzem ao longo do seu ciclo de vida
- O lado positivo disto: economia de recursos
- O lado menos-positivo disto: não é possível determinar um tamanho de área ideal e portanto prevenir erros de 'estouro' de área de alocação

TAD Pilha

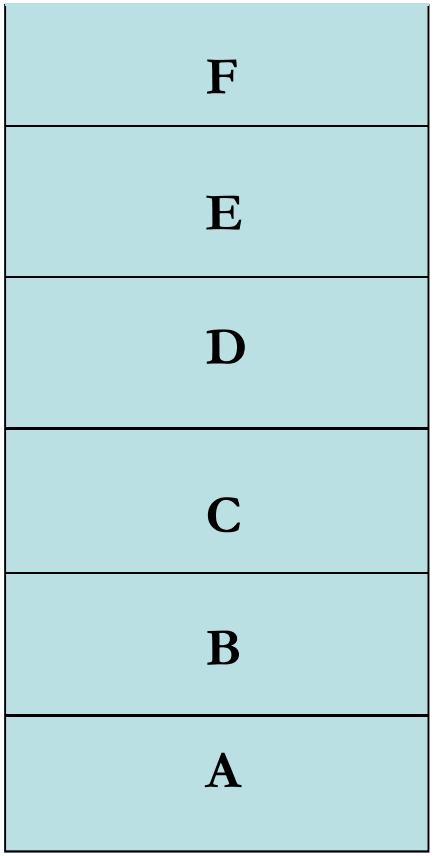
- Operações
  - Empilhar ou push
  - Desempilhar ou pop

Dada uma pilha **s** qualquer, as operações acima são especificadas como

i = s.pop(); leia-se: desempilhe de **s** e atribua para **i**

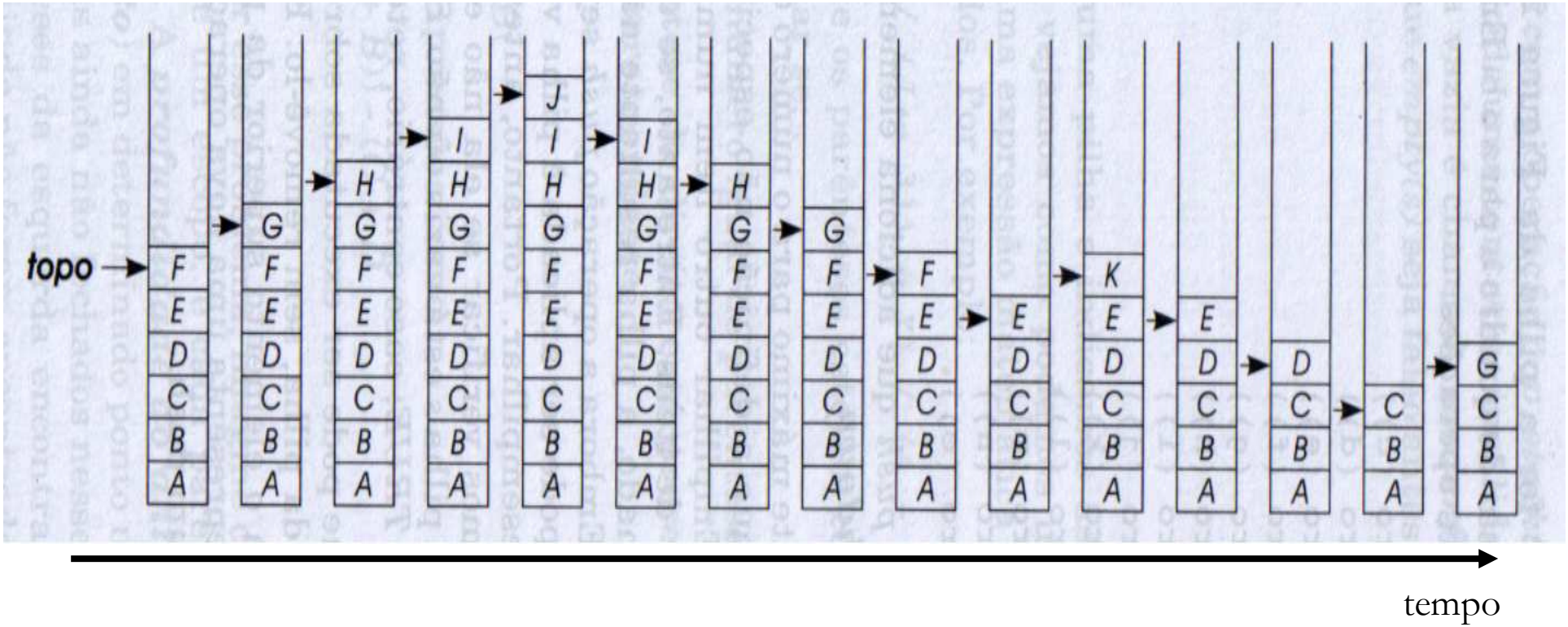
s.push(i); leia-se: empilhe **i** na pilha **s**

Exemplo: ciclo de vida de uma pilha




```
s.push (G);
s.push (H);
s.push (I);
s.push (J);
s.pop();
s.pop();
s.pop();
s.pop();
s.pop();
s.push(K);
s.pop();
s.pop();
s.pop();
s.push (G);
```

Exemplo: ciclo de vida de uma pilha



Operações - detalhes

- Não existe limite (teórico) para empilhar em s
  - Apenas limite imposto pelo meio físico de armazenamento
- Antes de desempilhar testar se a pilha é não vazia

  (s.isEmpty leia-se: cheque se a pilha s não é pilha vazia)
- Para apenas inpecionar o elemento do topo da pilha sem o remover, utilize a função top()
- i = s.top(); leia-se: leia o item no topo de s e atribua para i
  - Isto é equivalente à i = s.pop(); s.push(i);

Estudo de caso 1: avaliação de expressões

- Dada uma expressão aritmética qualquer:

  7 – ((X * ((X + Y) / (J – 3)) + Y) / (4 – 2.5))

- Condições de avaliação correta:

  1. Existe um número igual de parênteses esquerdos e direitos
  2. Todo parêntese da direita está precedido por um parêntese da esquerda correspondente

- Por definição estão erradas as sub-expressões
  - ((A+B ou A + B(=> viola condição 1
  -)A+B(-C ou (A+B)) – (-C+D => viola condição 2
- Pilhas podem ser usadas para checagem de parenteses em expressões

Estudo de caso 2: expressões pós-fixas

((4*5)+5)+6*5

- Na notação pós-fixa:

4 5 * 5 + 6 5 * +

- Usa-se pilha para <u>avaliar expressão pós-fixa</u>
  - Empilhar algarismos
  - A cada operador, desempilhar dois algarismos, fazer a operação e empilhar o resultado

Implementação de Pilhas como listas ligadas

- Definição
- Inicialização
- Pilha Vazia
- Pilha Cheia
- Empilhar (insert no início da lista)
- Desempilhar (remove do início)
- Consulta Topo (início da lista)

Pilhas como listas ligadas

```
public class Stack {
     private Node top;
     public Stack(){
       top = null;
     }
     ...
     public boolean isEmpty(){..}
     public void makeEmpty(){..}

     public void push(Object x){
       top = new Node(x,top);
     }
```

Pilhas como listas ligadas

```
  public Object getTop(){..}
  public Object pop throws
                UnderflowException{
    if (isEmpty())
      throw new UnderflowException();
    Object obj = top.getElem();
    top = top.getNext();
    return obj;
  }
}
```

Implementação de Pilhas como Arrays

- Evita referências
- Solução popular
  - Normalmente não há na prática pilhas com grande tamanho
- Implementação é mais simples
- Eficiência até melhor de listas ligadas
  - Porém mais testes são feitos

Pilhas como arrays

```
public class ArrayStack {
       private Object[] array;
       private int top;
       public ArrayStack(int tam){
         top = -1;
         array = new Object[tam];
       }
       ...
       public boolean isFull(){..}
       public boolean isEmpty(){..}
       public void makeEmpty(){
         this.top = -1;
         array = new Object[tam]; // garbage limpa!
       }

       public void push(Object x) throws
                              OverflowException{
         if (isFull()) throw new OverflowException();
         top++;
         array[top] = x;
       }
```

Pilhas como arrays

```
  public Object getTop(){..}
  public Object pop throws
                UnderflowException{
    if (isEmpty())
      throw new UnderflowException();
    Object x = array[top];
    array[top--] = NULL;
    return x;
  }
}
```

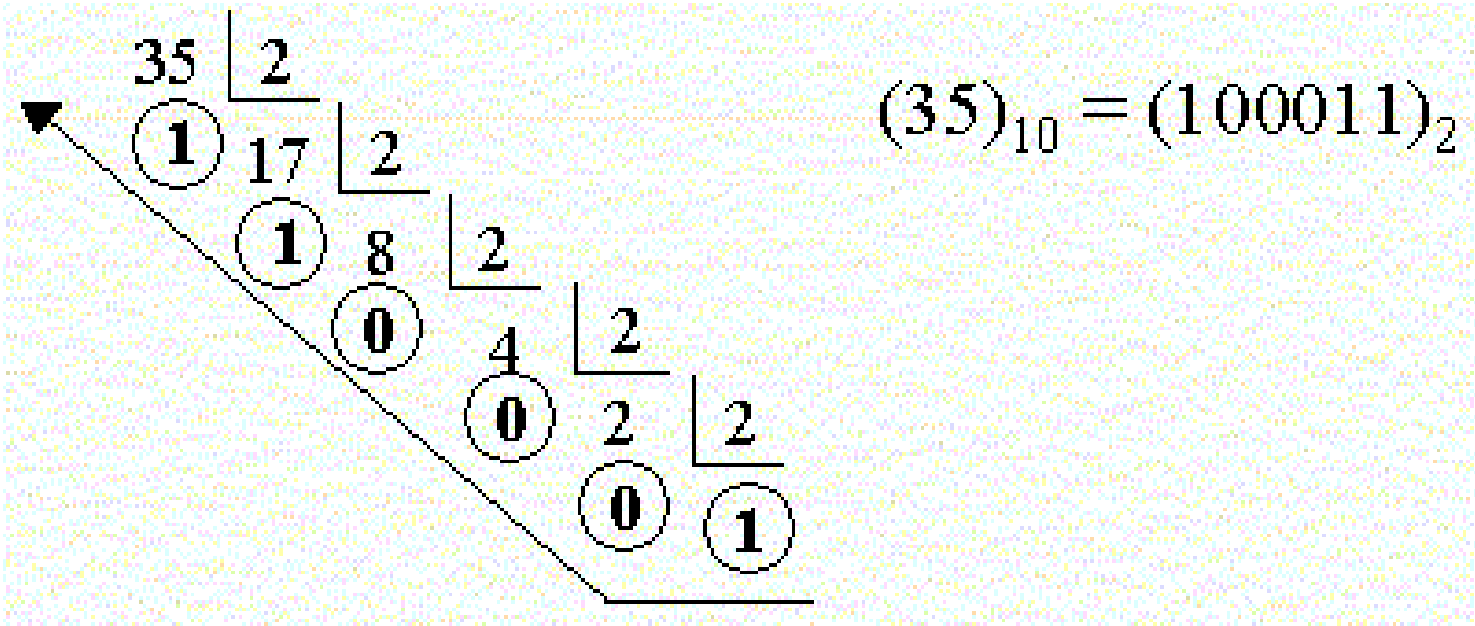
Exercício

- Escrever a função de conversão em Java:

$(35)_{10} \rightarrow (X)_2$, Quanto vale X ?

Exercício

- Conversão



Exercício

- Possibilidade de se utilizar pilhas