

Pilhas

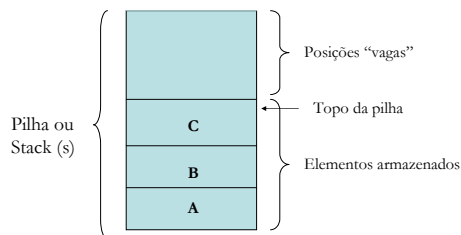
Prof. Tiago Massoni
Prof. Fernando Buarque
Engenharia da Computação
Poli - UPE

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”

2

Intuição



3

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

4

TAD Pilha

• Operações

- Empilhar ou push
- Desempilhar ou pop

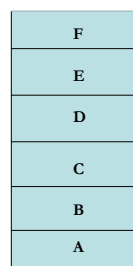
Dada uma pilha *s* qualquer, as operações acima são especificadas como

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

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

5

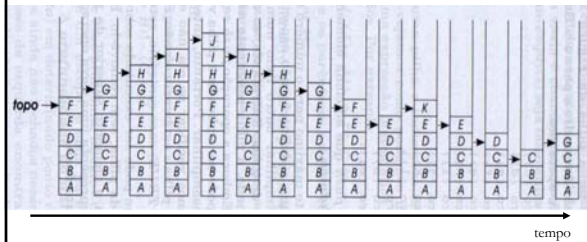
Exemplo: ciclo de vida de uma pilha



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

6

Exemplo: ciclo de vida de uma pilha



7

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 inspecionar 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 $i = s.pop()$; $s.push(i)$;

8

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:
 - Existe um número igual de parênteses esquerdos e direitos
 - Todo parêntese da direita está precedido por um parêntese da esquerda correspondente
- Por definição estão erradas as sub-expressões
 - $((A+B$ ou $A + B($ \Rightarrow viola condição 1
 - $)A+B(-C$ ou $(A+B)) - (-C+D$ \Rightarrow viola condição 2
- Pilhas podem ser usadas para checagem de parenteses em expressões

9

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 avaliar expressão pós-fixa
 - Empilhar algarismos
 - A cada operador, desempilhar dois algarismos, fazer a operação e empilhar o resultado

10

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)

11

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);
    }
}
```

12

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;
}
```

13

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

14

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;
    }

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

15

Pilhas como arrays

```
public Object getTop(){..}
public Object pop throws
    UnderflowException{
    if (isEmpty())
        throw new UnderflowException();
    return array[top - 1];
}
```

16

Exercício

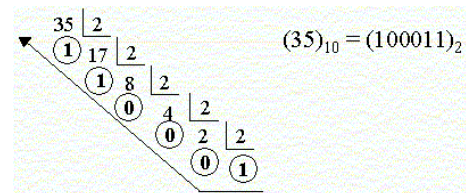
- Escrever a função de conversão em Java:

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

17

Exercício

- Conversão



18

Exercício

- Possibilidade de se utilizar pilhas

