

ALGORITMOS E ESTRUTURAS DE DADOS I

Prof. Caio César de Freitas Dantas

Filas

- Tipo Abstrato de Dados com a seguinte característica:
 - O primeiro elemento a ser inserido é o primeiro a ser retirado.
 - FIFO - First in First Out.
- Analogia: fila bancária, fila do cinema, etc.
- Usos: Sistemas operacionais: fila de impressão, fila de processamento, etc.

Pilhas

- É uma lista linear em que todas as inserções, retiradas e, geralmente, todos os acessos são feitos em apenas um extremo da lista.
- Os itens são colocados um sobre o outro. O item inserido mais recentemente está no topo e o inserido menos recentemente no fundo.
- O modelo intuitivo é o de um monte de pratos em uma prateleira, sendo conveniente retirar ou adicionar pratos na parte superior.

Pilhas

- Propriedade: o último item inserido é o primeiro item que pode ser retirado da lista. São chamadas listas lifo (“last-in, first-out”).
- Existe uma ordem linear para pilhas, do “mais recente para o menos recente”.
- É ideal para processamento de estruturas aninhadas de profundidade imprevisível.
- Uma pilha contém uma sequência de obrigações adiadas. A ordem de remoção garante que as estruturas mais internas serão processadas antes das mais externas.

Pilhas

- Aplicações em estruturas aninhadas:
 - Quando é necessário caminhar em um conjunto de dados e guardar uma lista de coisas a fazer posteriormente.
 - O controle de sequências de chamadas de subprogramas.
 - A sintaxe de expressões aritméticas.

Pilhas

- Uma pilha é uma estrutura de dados em que todo o acesso a seus elementos é feito através do seu topo.

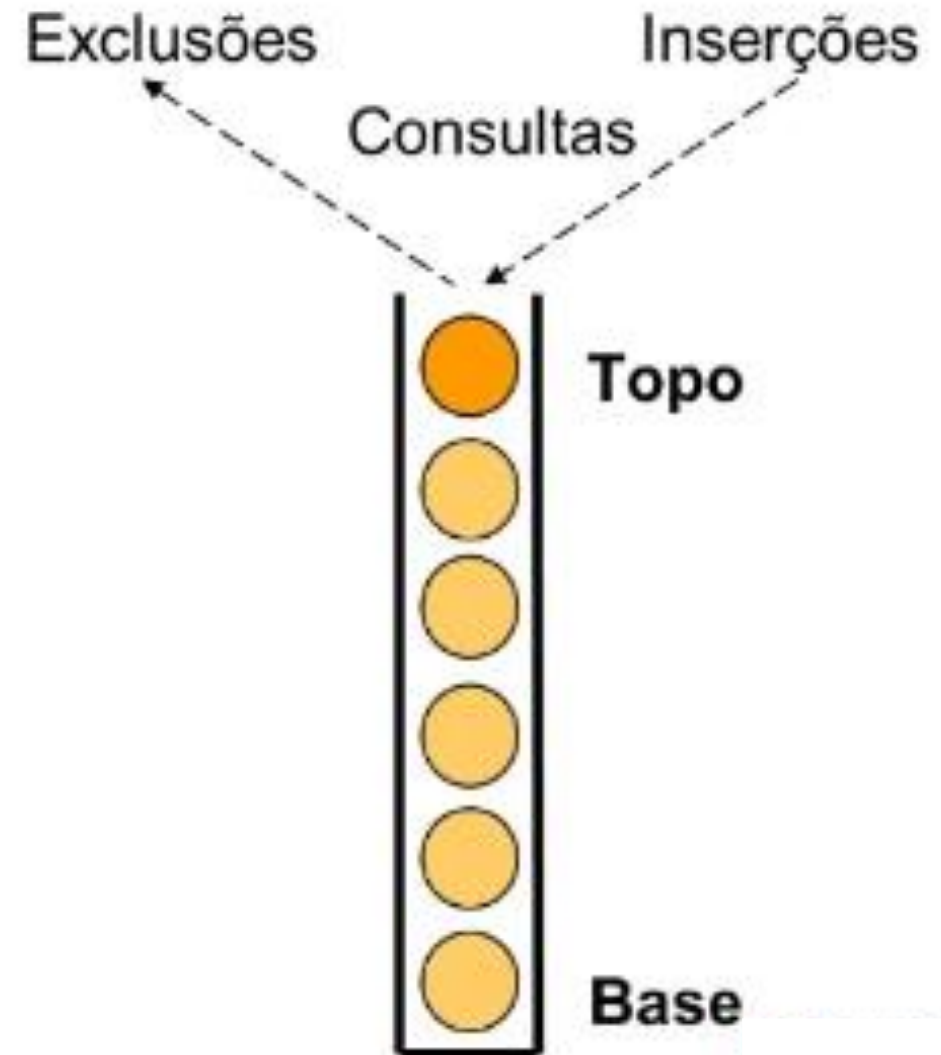
Operações Básicas

- Criar pilha;
- Inserção de um elemento do topo;
- Remoção de um elemento do topo;
- Acesso ao elemento do topo;
- Destruir pilha;

Pilhas

Características

- O elemento removido é o que esta na estrutura há menos tempo.
- O ultimo objeto inserido na pilha é também o primeiro a ser removido.



Pilhas

Índices de controle da pilha

Limite

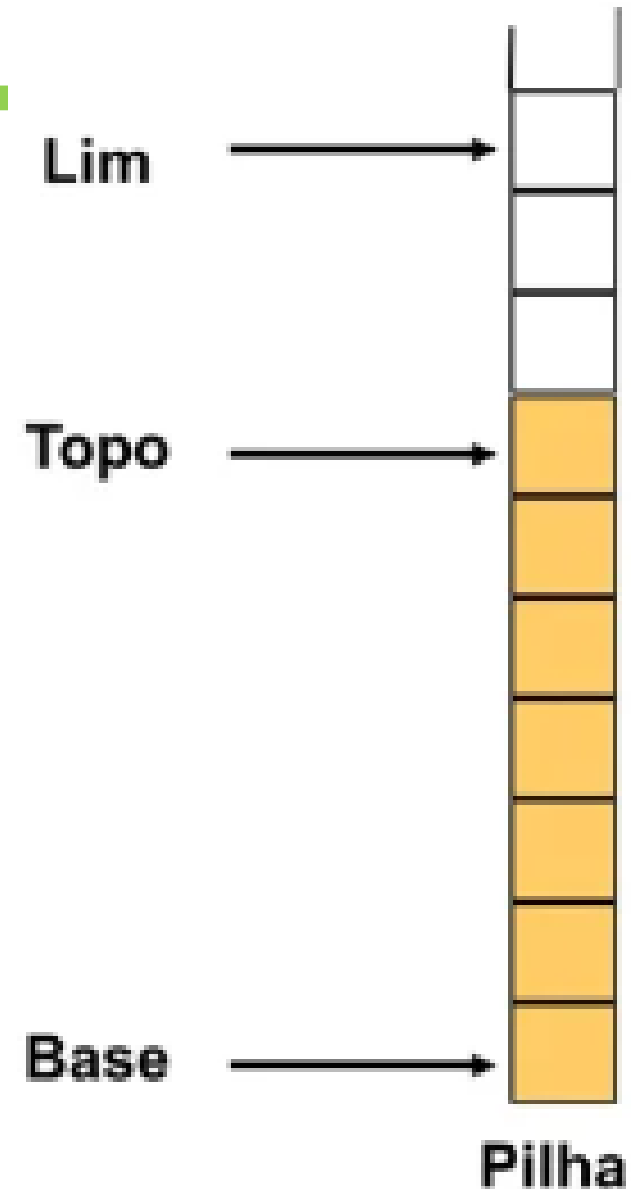
- Tamanho máximo que pode ser ocupado

Topo

- Atual posição de consulta da pilha

Base

- Ponto de inicio da pilha



Pilhas

- Executando as operações na pilha

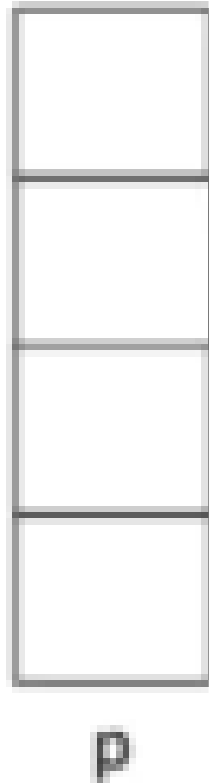


Qual será o estado final da pilha após estas operações ?

```
p = cria_pilha()  
empilha (p, 10)  
empilha (p, 20)  
empilha (p, 30)  
desempilha (p)  
empilha (p, 50)  
desempilha (p)  
desempilha (p)  
empilha (p, 90)
```

Pilhas

- Executando as operações na pilha



→ `p = cria_pilha ()`
→ `empilha (p, 10)`
→ `empilha (p, 20)`
→ `empilha (p, 30)`
→ `desempilha (p)`
→ `empilha (p, 50)`
→ `desempilha (p)`
→ `desempilha (p)`
→ `empilha (p, 90)`

Pilhas

- Aplicações com pilhas

Identificando palíndromos com pilha

- Palíndromos são palavras/frases que são iguais quando lidas de frente para trás

Exemplos:

- Ana
- Arara
- Rotor
- Socorram me subi no onibus em marrocos
 - Desconsiderando os espaços em branco

Pilhas

Algoritmo

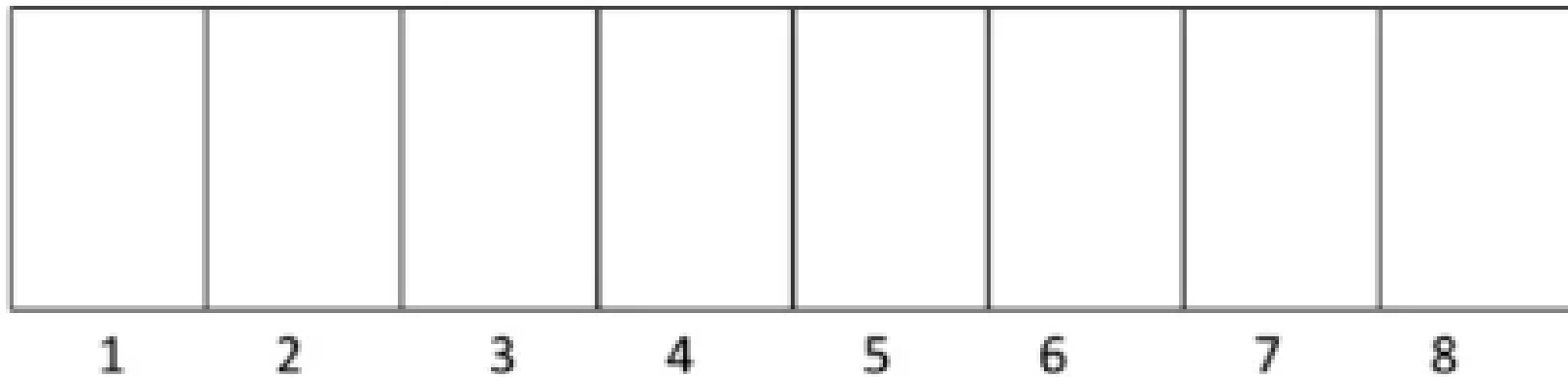
1. empilhe $n/2$ letras
2. descarta letra central se houver
3. repita
 1. compare o topo da pilha com a proxima letra
 2. se são iguais entao desempilhe
 3. senao não é um palindromo
4. se no final a pilha esta vazio entao a palavra é um palindromo

Pilhas

ARARA

Topo = 0

Base = 1

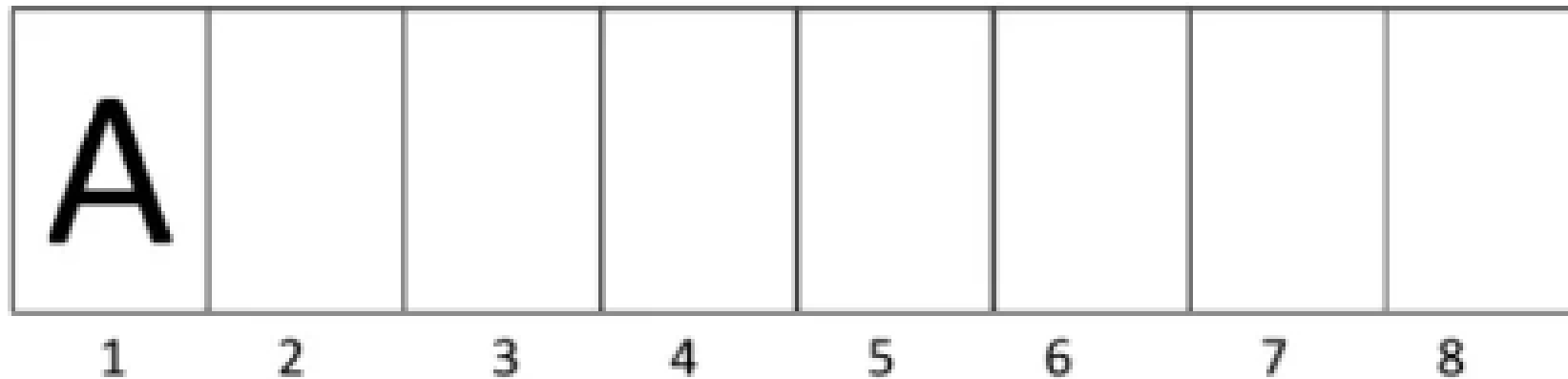


Pilhas

ARARA

Topo = 1

Base = 1



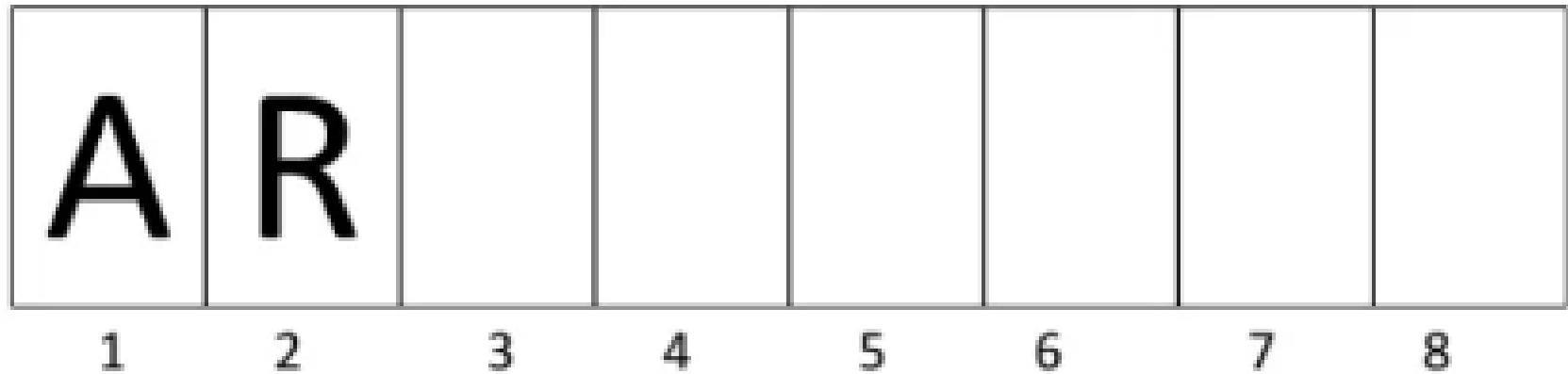
Empilha letra 1

Pilhas

ARARA

Topo = 2

Base = 1



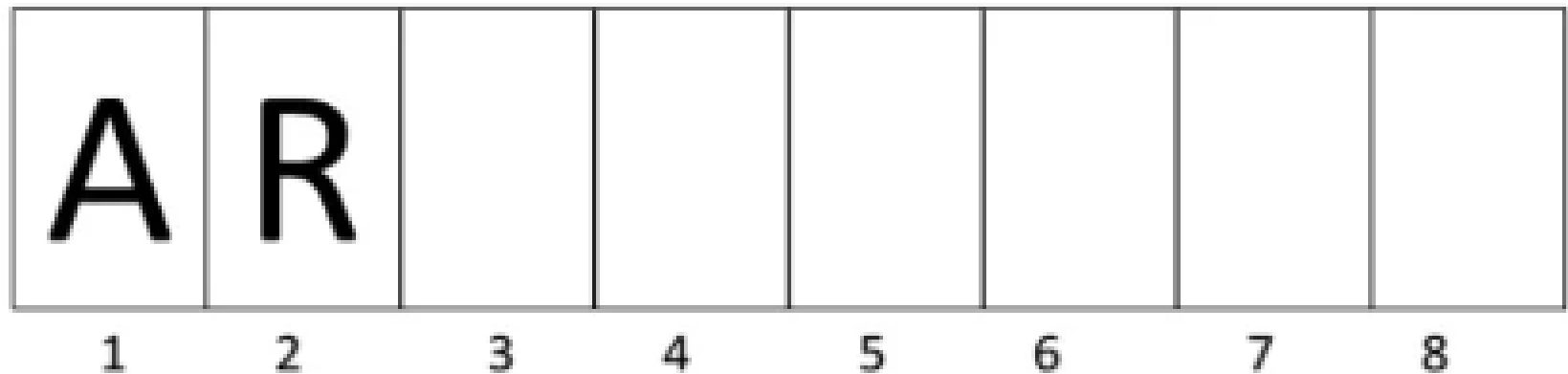
Empilha letra 2

Pilhas

ARARA

Topo = 2

Base = 1



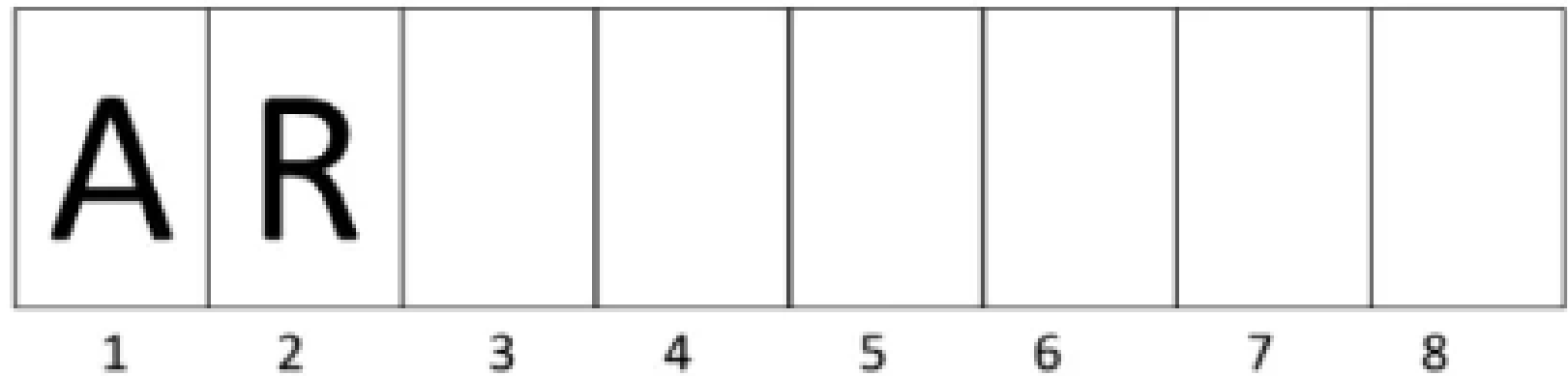
Descarta letra central

Pilhas

ARA**RA**

Topo = 2

Base = 1



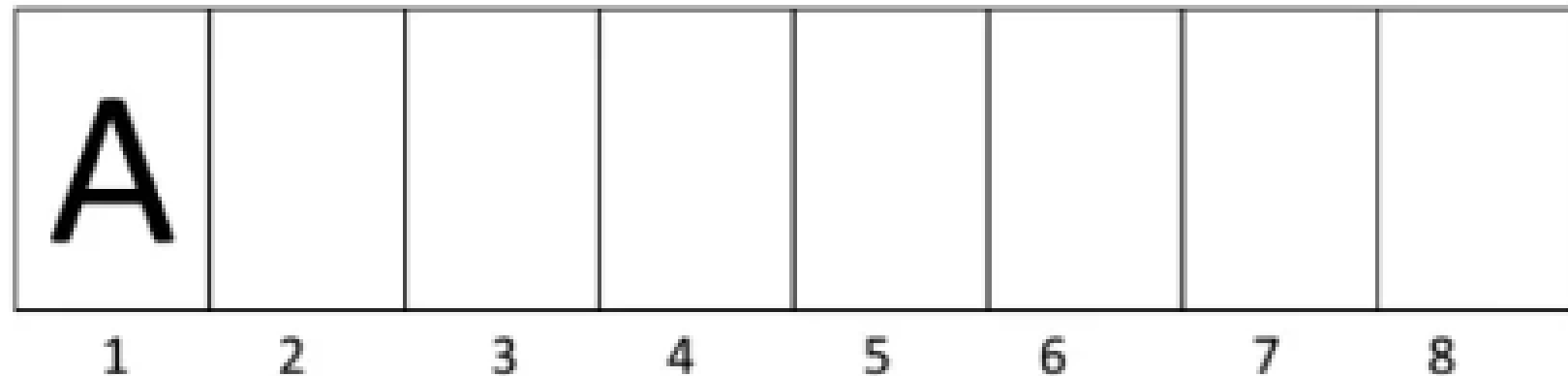
Comparar topo com próxima letra

Pilhas

ARARA

Topo = 1

Base = 1



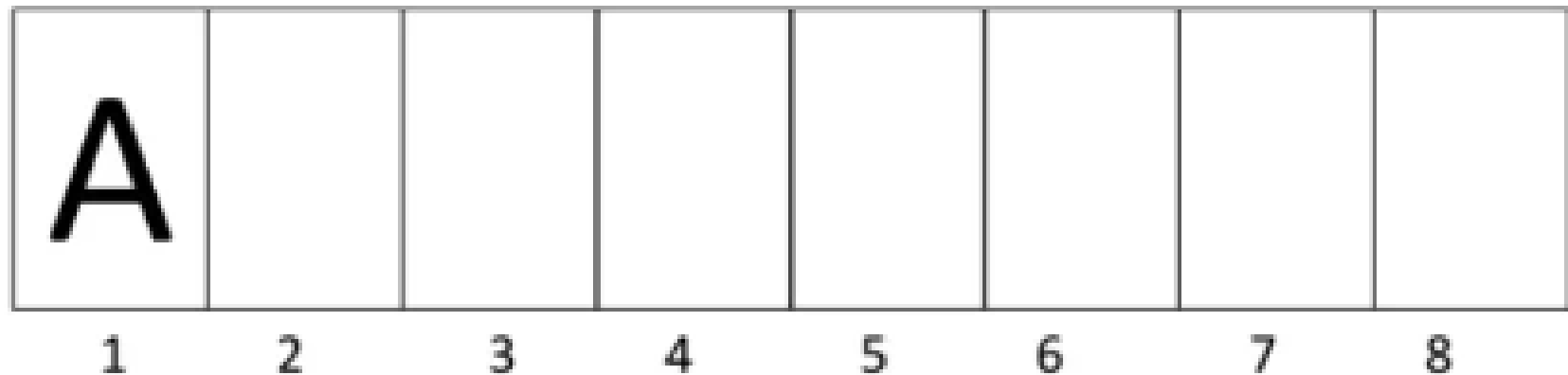
Se são iguais então desempilhe

Pilhas

ARARA

Topo = 1

Base = 1



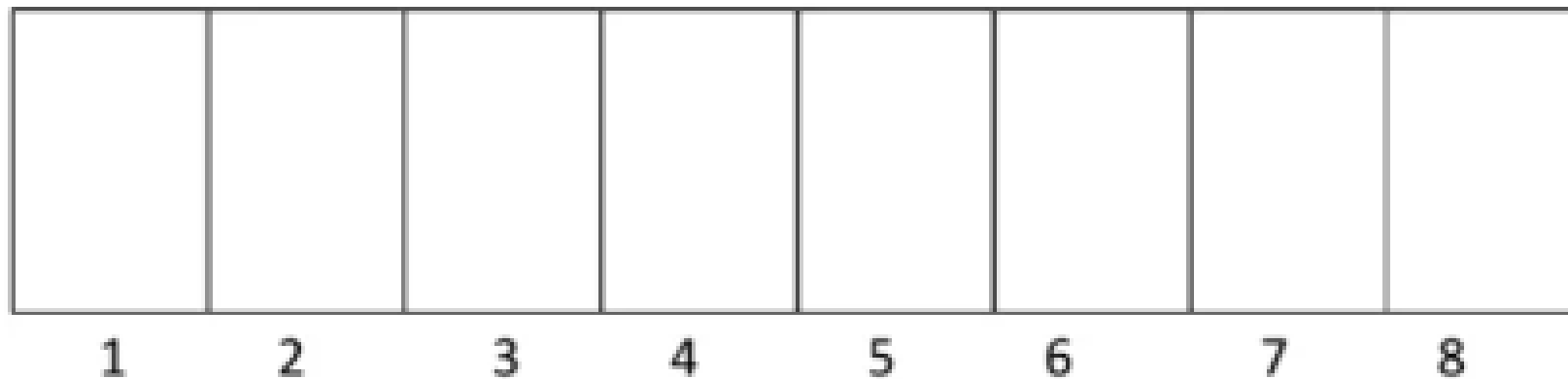
Comparar topo com próxima letra

Pilhas

Topo = 1

Base = 1

Se no final a pilha está vazia então é um palíndromo



Se são iguais então desempilhe

Pilhas

Pilhas podem ser utilizadas para verificar se os parênteses em uma expressão estão balanceados

Exemplo

$((3+4+(4*9)$

Falta dois parênteses fechando!

Pilhas

Para cada carácter da expressão faça

- Se encontrou um "(" empilha
- Se encontrou um ")" então
 - Se a pilha estiver vazia: expressão invalida
 - Caso contrario desempilhe
- Se no final pilha estiver vazia, expressão válida.

$((3 + 4 + (4 * 9))$

Pilhas

Implementação do TAD Pilha

- Existem várias opções de estruturas de dados que podem ser usadas para representar pilhas.
- As duas representações mais utilizadas são as implementações por meio de vetores e de estruturas encadeadas.

Pilhas

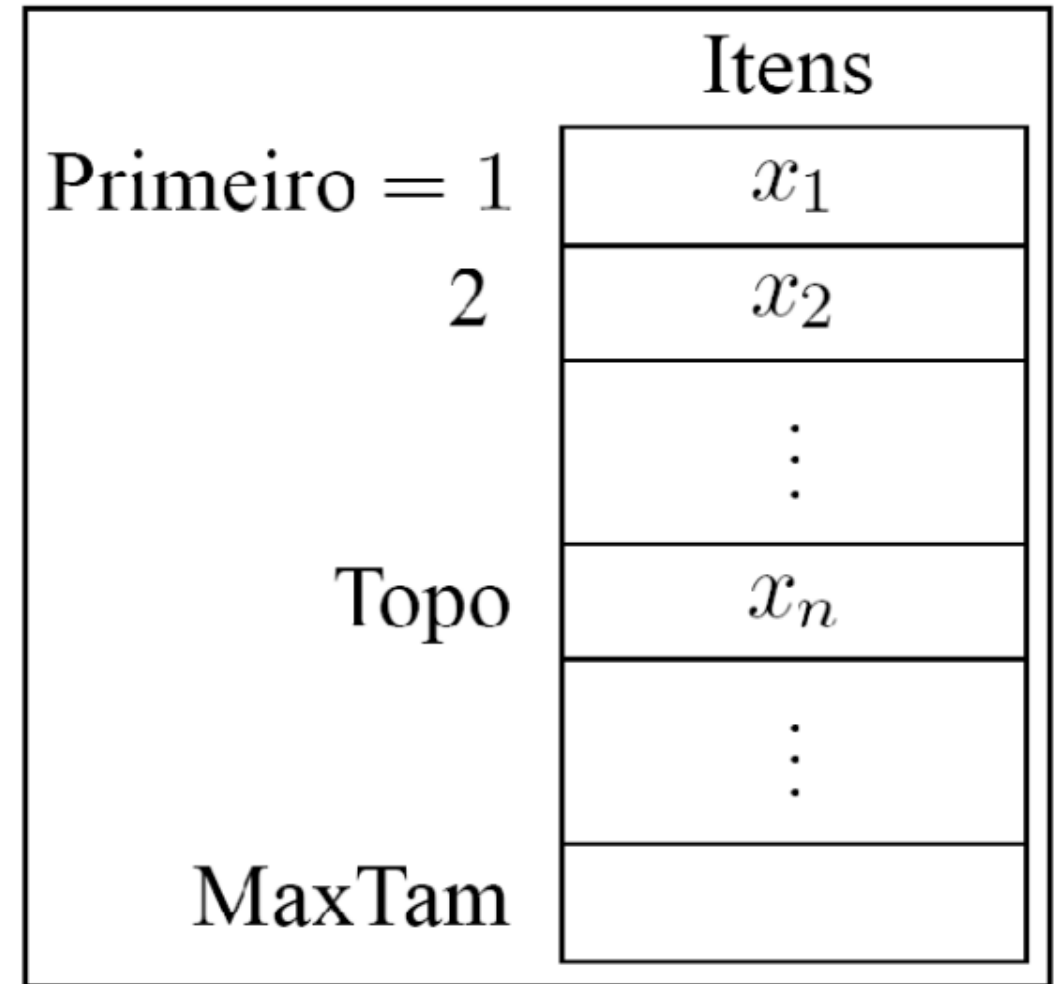
Pilhas em Alocação Sequencial e Estática

- Os itens da pilha são armazenados em posições contíguas de memória.
- Como as inserções e as retiradas ocorrem no topo da pilha, um cursor chamado Topo é utilizado para controlar a posição do item no topo da pilha.

Pilhas

Pilhas em Alocação Sequencial e Estática

- Os itens são armazenados em um vetor de tamanho suficiente para conter a pilha.
- O outro campo do mesmo registro contém o índice do item no topo da pilha.
- A constante MaxTam define o tamanho máximo permitido para a pilha.

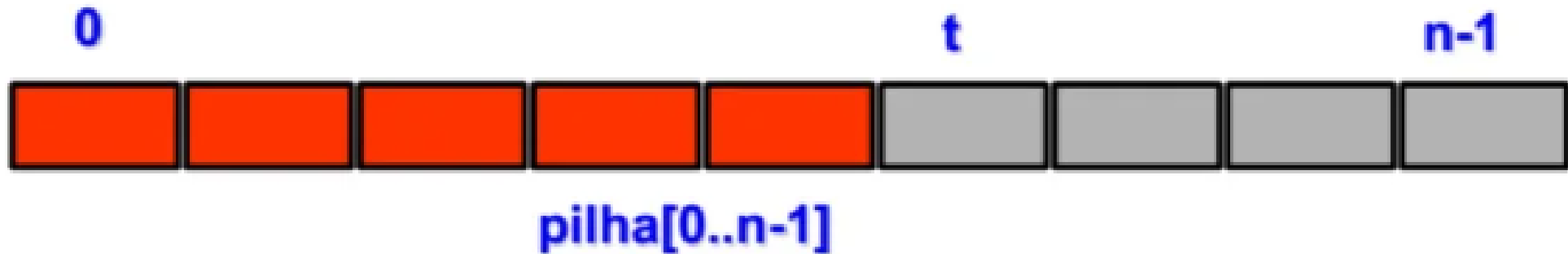


Pilhas

Supondo que a pilha está armazenada em um vetor

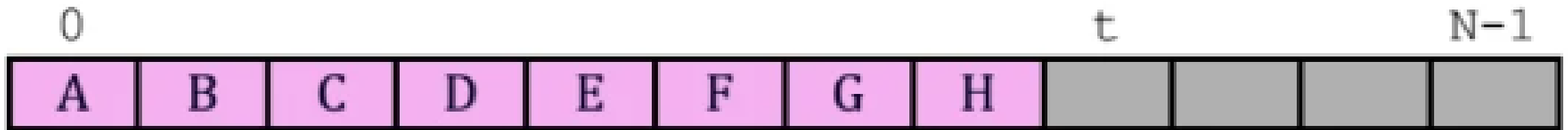
$\text{pilha}[0..n-1]$

A parte do vetor ocupada pela pilha será:



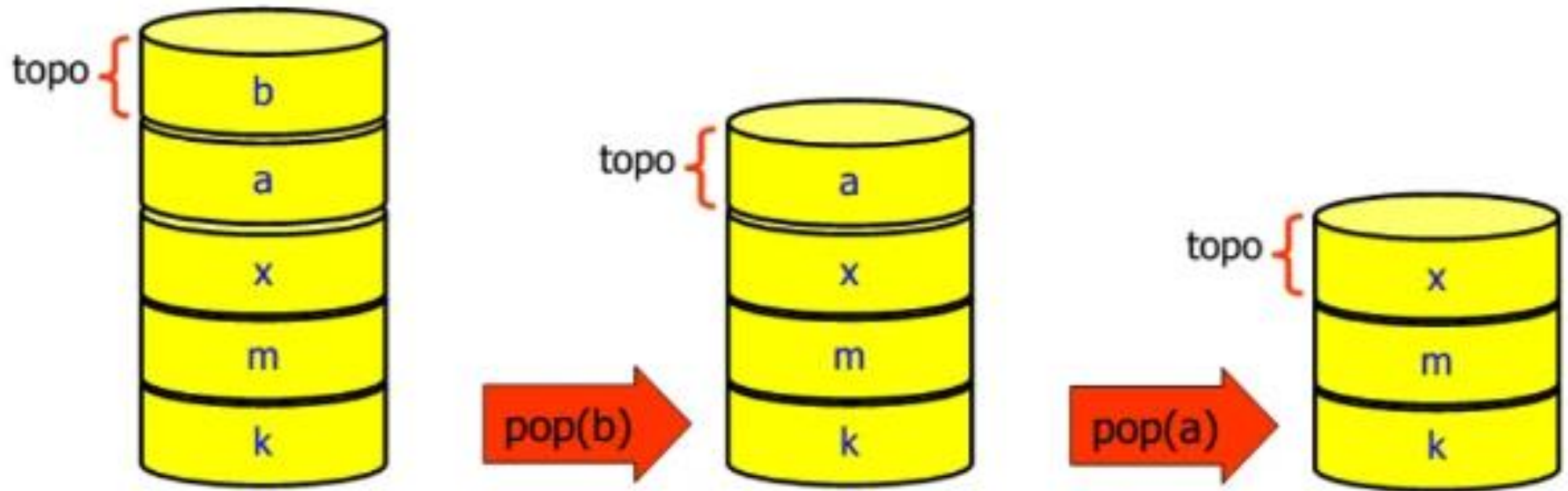
Pilhas

- A natureza dos elementos do vetor é irrelevante, eles podem ser inteiros, caracteres, etc..
- O índice t indica a primeira da pilha.
- A pilha está vazia se $t=0$ e cheia se $t = N$



Pilhas

- Removendo um elemento



Pilhas

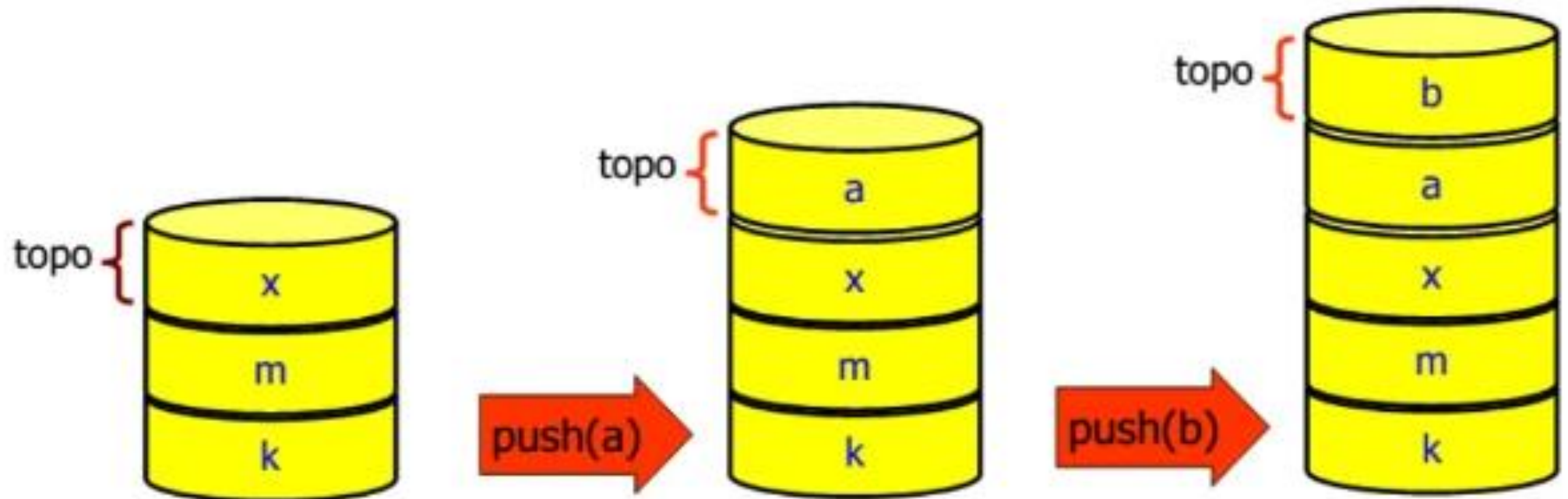
- Para remover (desempilhar, pop) um elemento:

```
char desempilha (void) {  
    return pilha [--t];  
}
```

```
t -= 1;  
x = pilha[t];  
x = pilha[--t];
```

Pilhas

- Inserindo um elemento



Pilhas

- Para inserir (empilhar, push) o elemento y na pilha:

```
void empilha (char y) {  
    pilha [t++] = y;  
}
```

```
pilha[t] = y;  
t += 1;  
pilha[t++] = y;
```

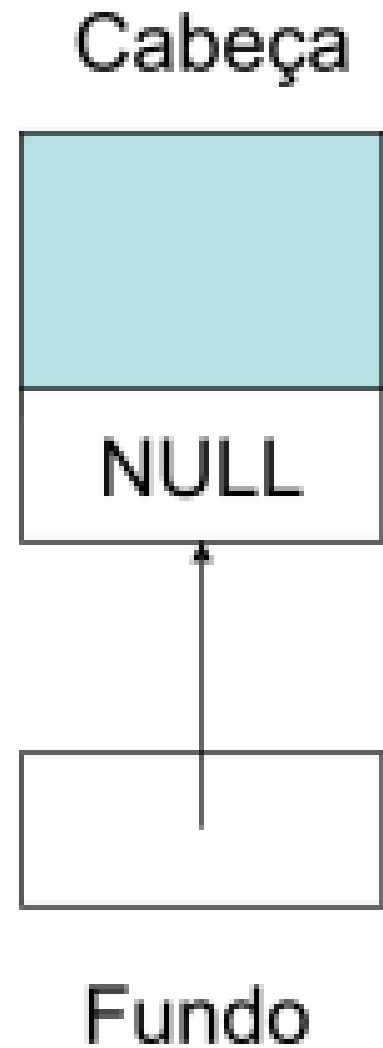
Pilhas

Implementação por PONTEIRO

- Criar um campo tamanho evita a contagem do número de itens na função tamanho.
- Cada célula de uma pilha contém um item da pilha e um apontador para outra célula.
- O registro (struct) TPilha contém um apontador para o topo da pilha (célula cabeça) e um apontador para o fundo da pilha.
 - Funcionam como início e fim de uma lista

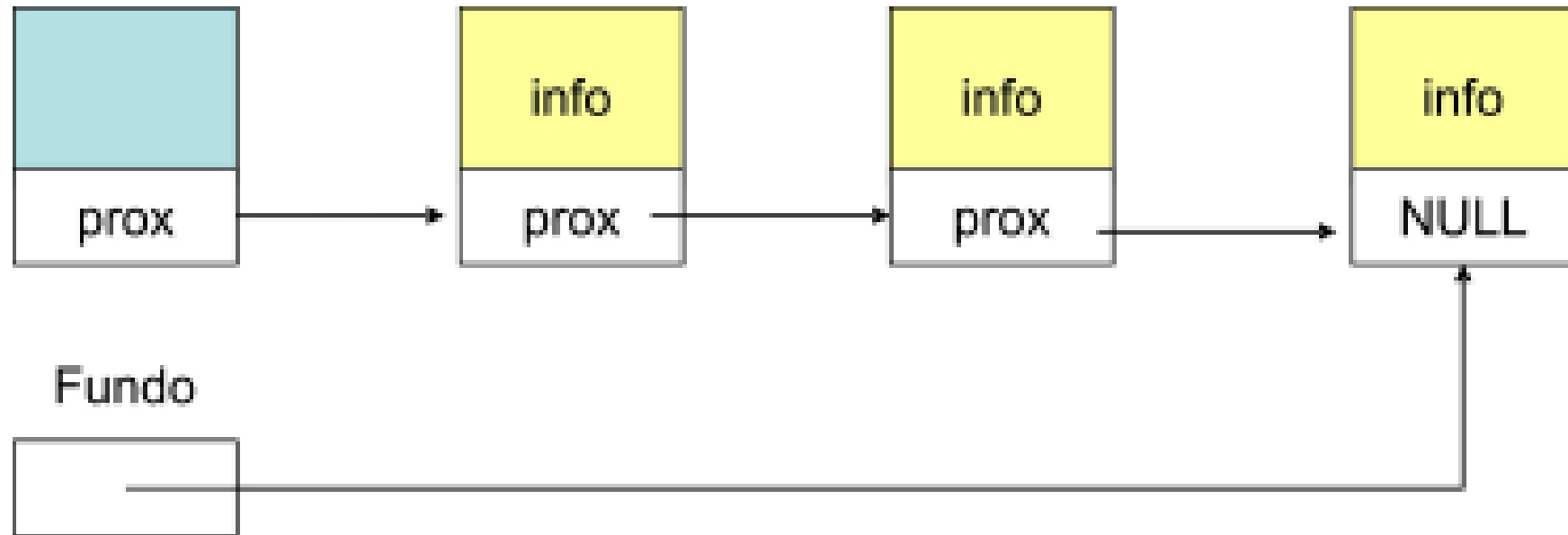
Pilhas

Criar Pilha Vazia (usando célula Cabeça)



Pilhas

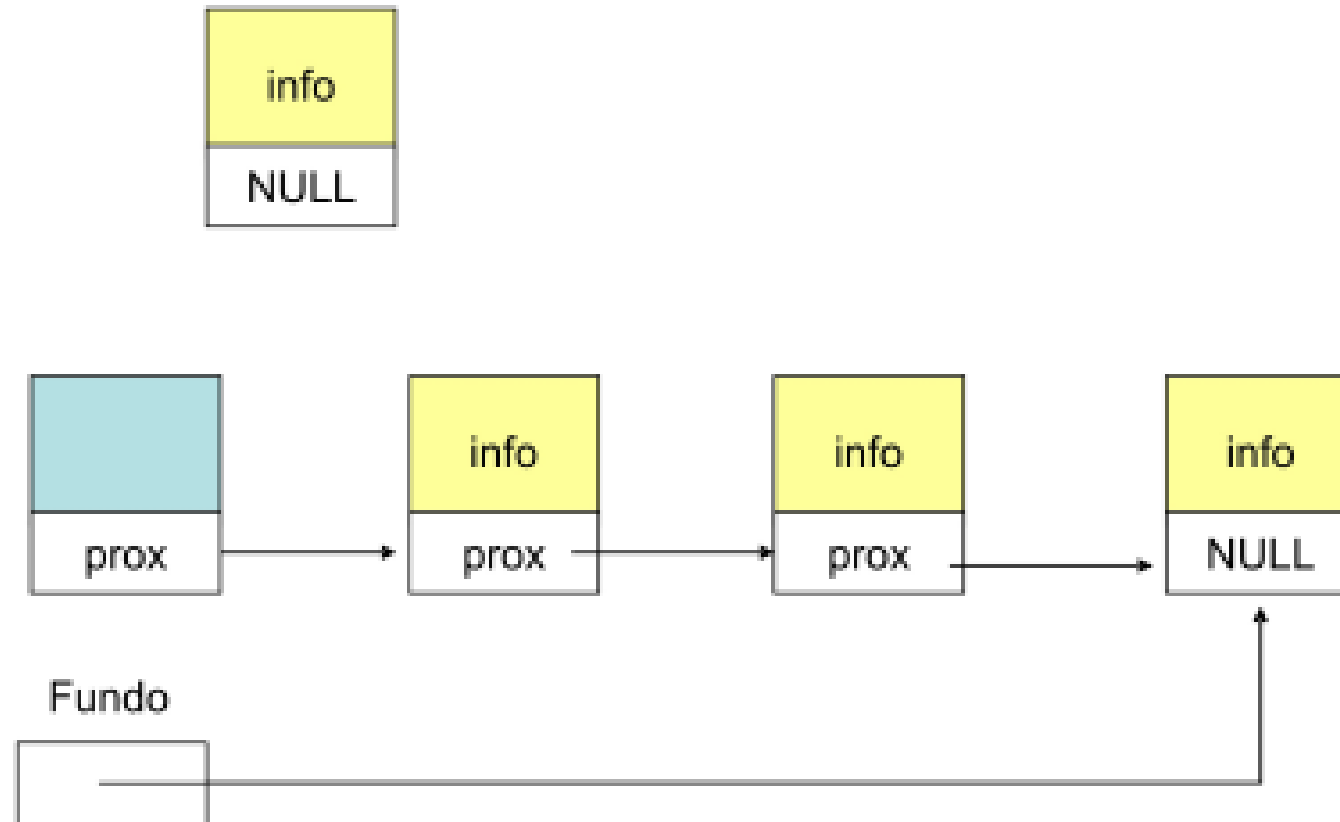
Inserção de Novos Elementos



Opção única de posição onde se pode inserir: Topo da pilha, ou seja, primeira posição

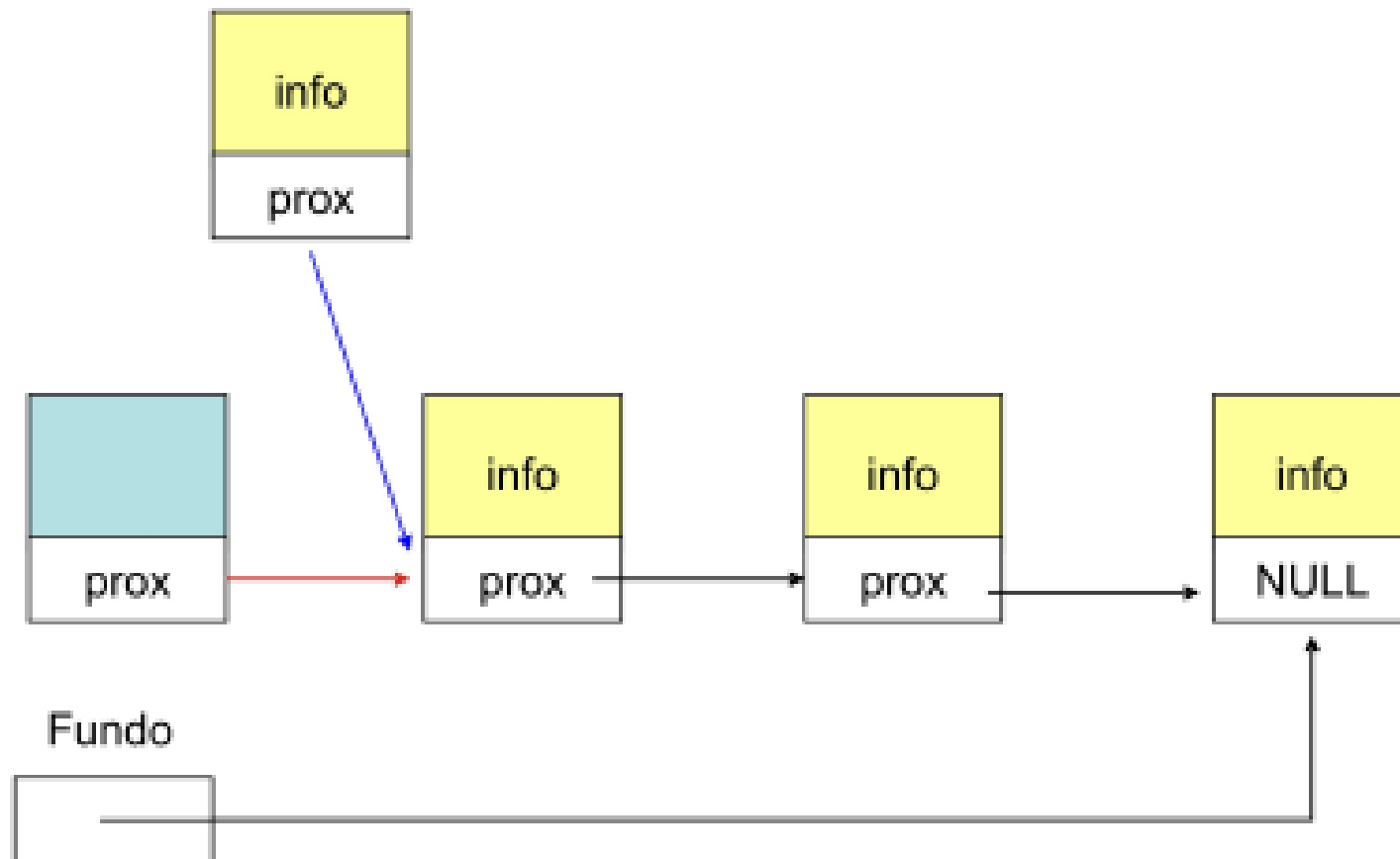
Pilhas

Inserção de Novos Elementos



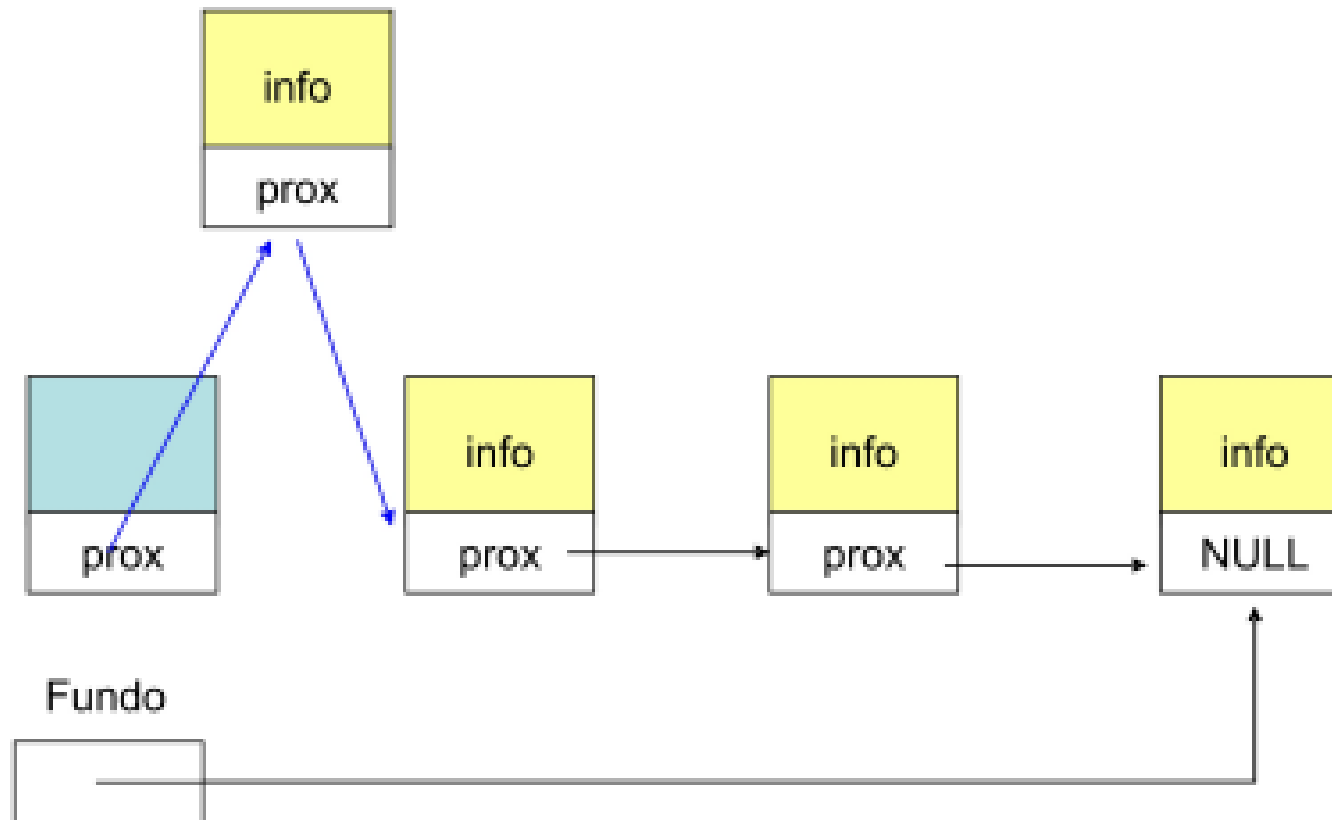
Pilhas

Inserção de Novos Elementos



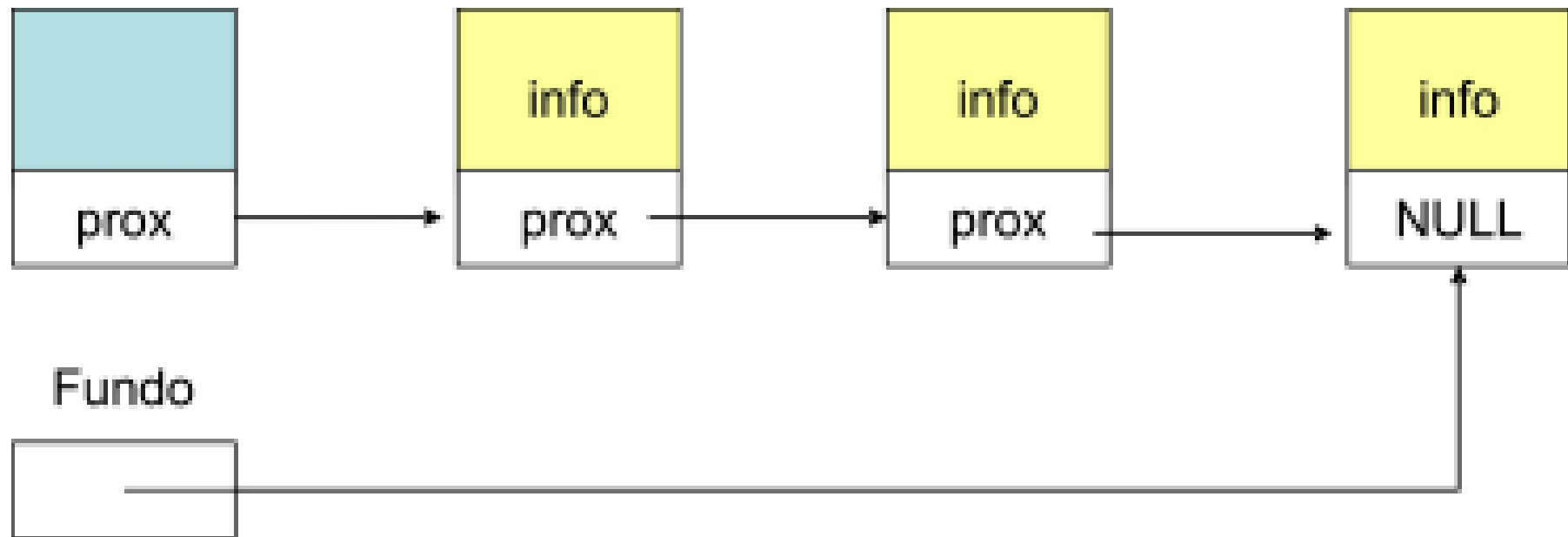
Pilhas

Inserção de Novos Elementos



Pilhas

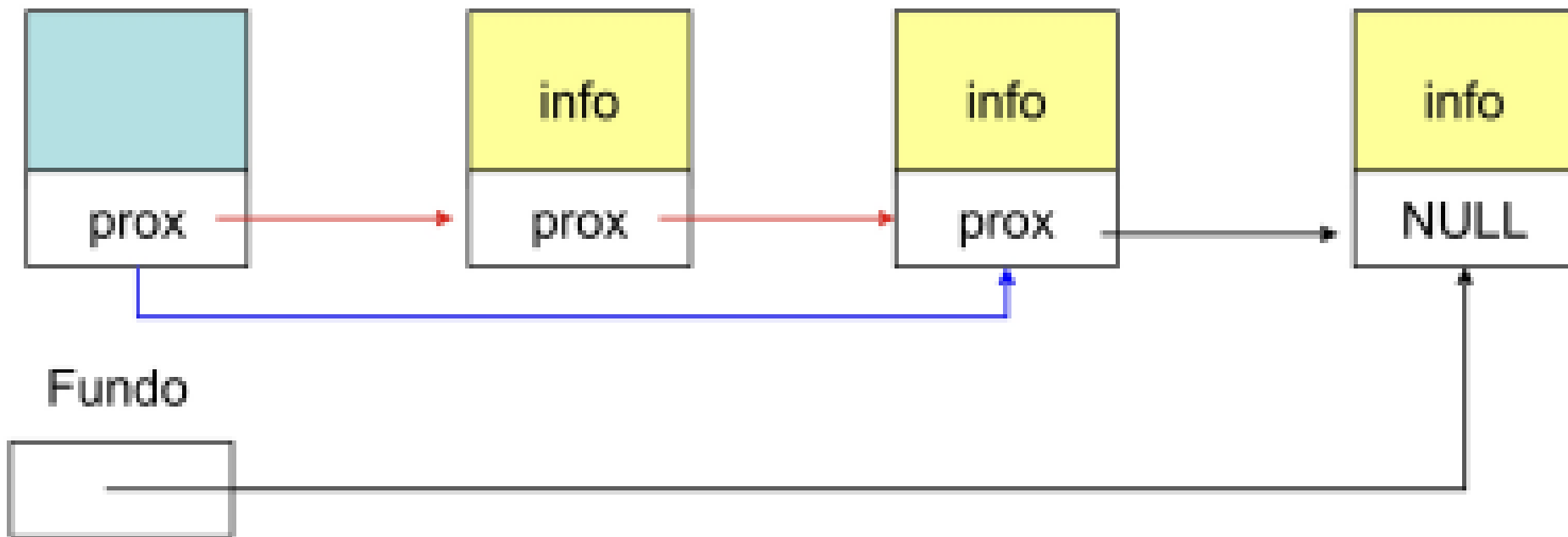
Retirada de Elementos



Opção única de posição onde se pode retirar: Topo da pilha, ou seja, primeira posição

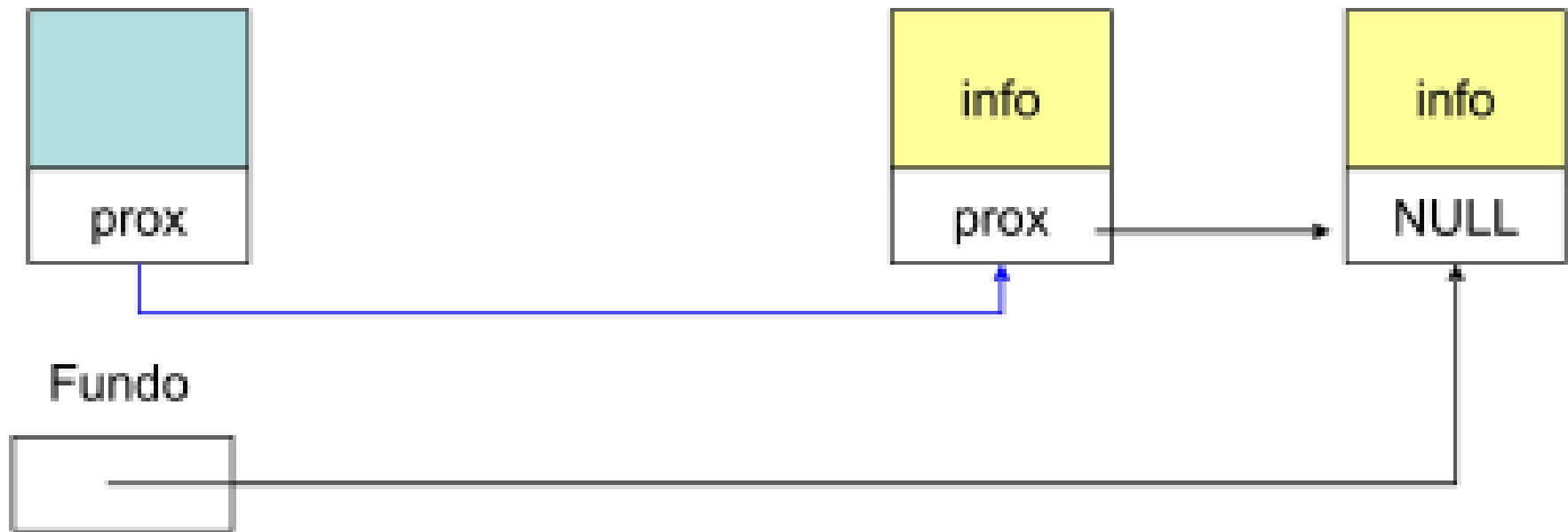
Pilhas

Retirada de Elementos



Pilhas

Retirada de Elementos



Pilhas

Implementações com Vetor (Array)

```
void criapilha (void) {  
    t = 0;  
}
```

```
void empilha (char y) {  
    pilha[t++] = y;  
}
```

```
char desempilha (void) {  
    return pilha[--t];  
}
```

```
int pilhavazia (void) {  
    return t <= 0;  
}
```

FIM!

A thick horizontal green line spans the width of the slide, and a thick diagonal green line runs from the bottom-left corner towards the top-left, meeting the horizontal line.