

# PILHA

## Algoritmos e Programação II

(slides baseados no material do prof. Fábio Viduani)

# INTRODUÇÃO

- Lista linear especial
- Política de inserções e remoções bem definida
- Inserção e remoção são as únicas operações

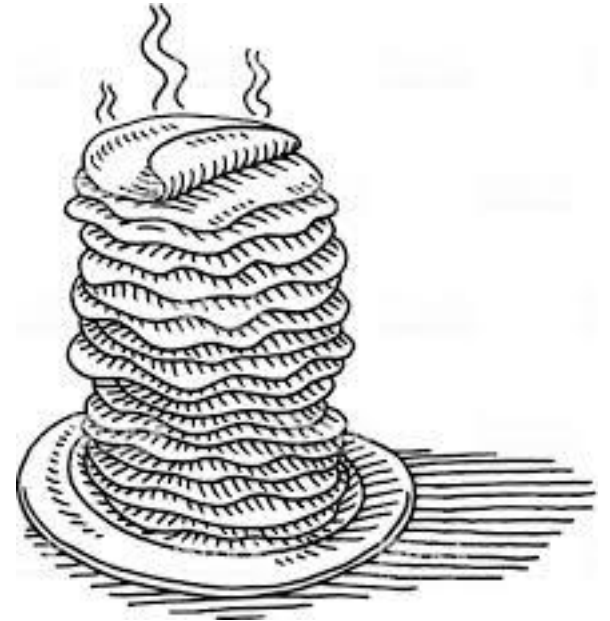
# DEFINIÇÃO

- PILHA é uma lista linear tal que as operações de inserção e remoção são realizadas em **um único extremo** da lista.
- Funcionamento dessa estrutura pode ser comparado a qualquer pilha de pratos de um Restaurante, por exemplo. Os funcionários colocam pratos limpos também nesse mesmo ponto da pilha. Seria estranho ter de movimentar pratos, por exemplo, no meio ou no final da pilha, sempre que uma dessas operações fosse realizada.

# DEFINIÇÃO

- Indicador do extremo onde ocorrem as operações de inserção e remoção: chamado de **TOPO** da pilha.
- Tais operações são chamadas de **empilhamento (empilhar)** e **desempilhamento (desempilhar)**.
- A operação de busca não foi mencionada e não faz parte do conjunto de operações básicas de uma pilha.







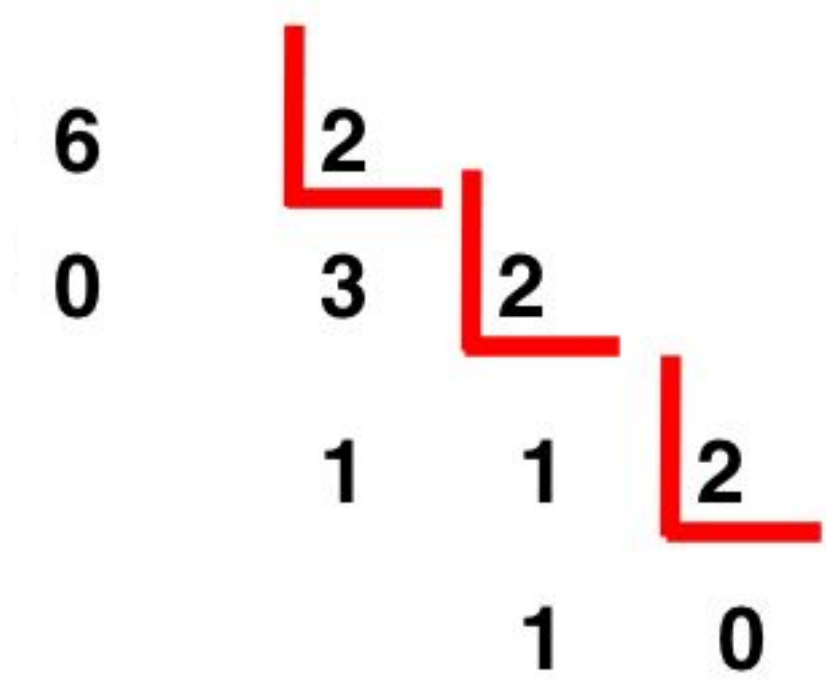
# APLICAÇÕES

- Utiliza-se pilha em aplicações em que os dados são obtidos na ordem inversa àquela em que foram fornecidos.

Exemplo:

- Conversão de número decimal para binário;
- Retirada de mercadorias de um caminhão de entregas;
- Mecanismo de fazer/desfazer do Word;
- Mecanismo de navegação de páginas na Internet  
(avançar e retornar).





110

# DEFINIÇÃO

- Considere que as células da pilha são do tipo abaixo:

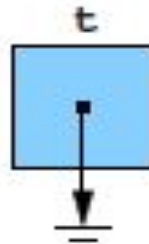
```
struct celula {  
    int chave;  
    struct celula *prox;  
};
```

# DEFINIÇÃO

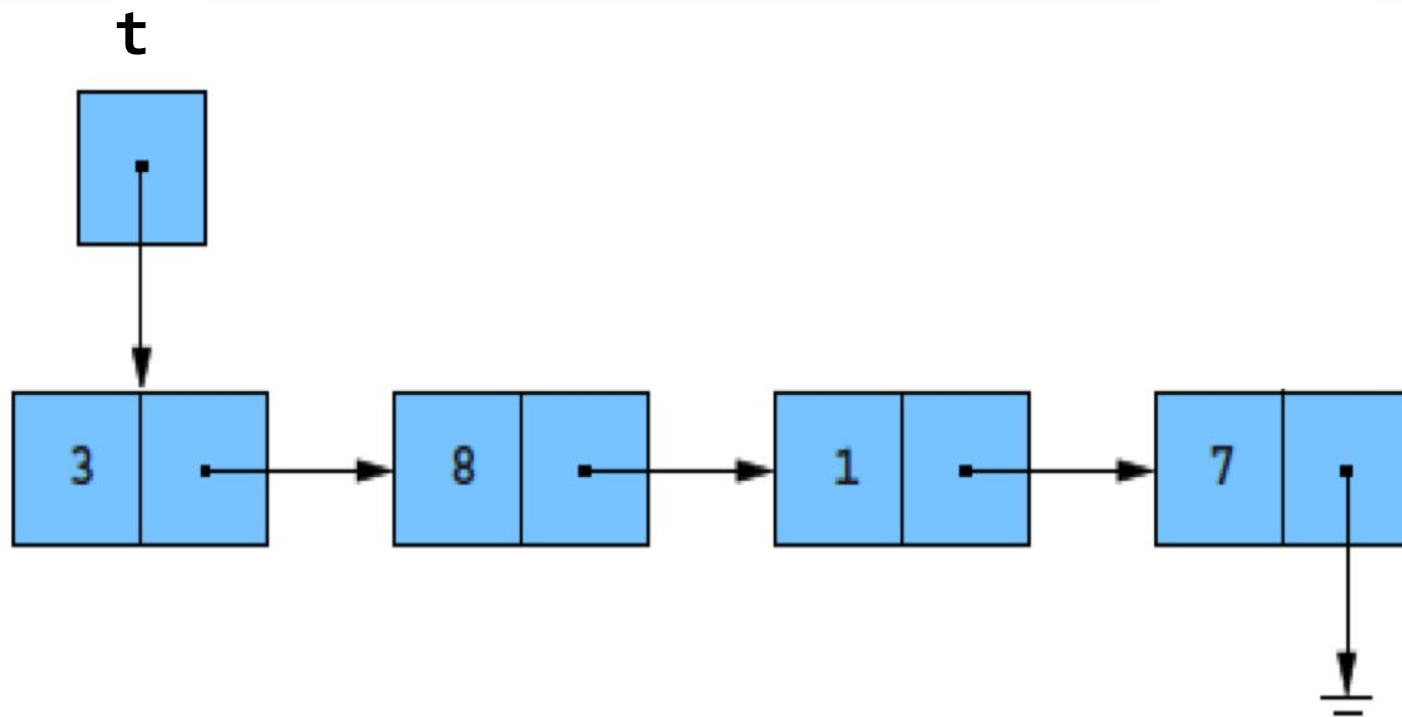
- A inicialização de uma pilha vazia em alocação encadeada é dada a seguir: PILHA  $\rightarrow$  1 extremo

```
celula *t;
```

```
t = NULL;
```



REPRESENTAÇÃO DE UMA **PILHA VAZIA** EM ALOCAÇÃO ENCADEADA

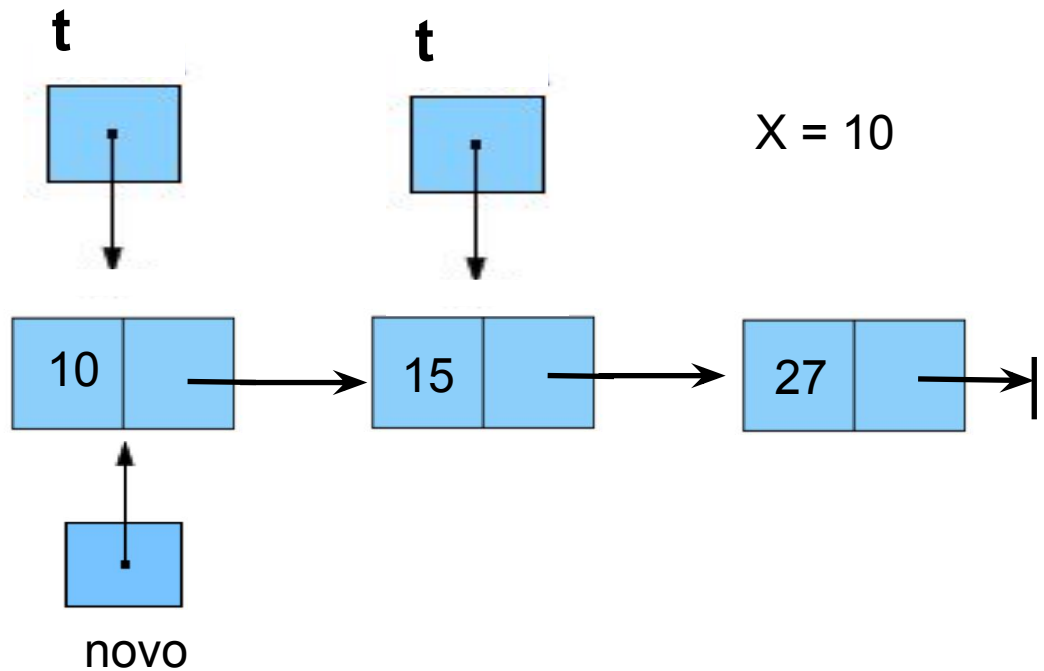


REPRESENTAÇÃO/EXEMPLO DE UMA **PILHA NÃO VAZIA** EM ALOCAÇÃO ENCADEADA

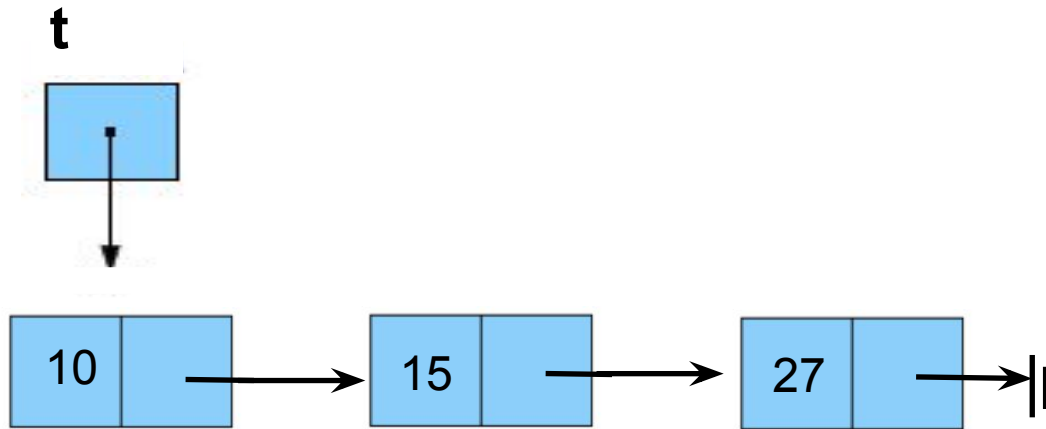
# OPERAÇÕES

- As operações básicas são:
  - empilhar
  - desempilhar

# OPERAÇÃO: empilhar



# OPERAÇÃO: empilhar

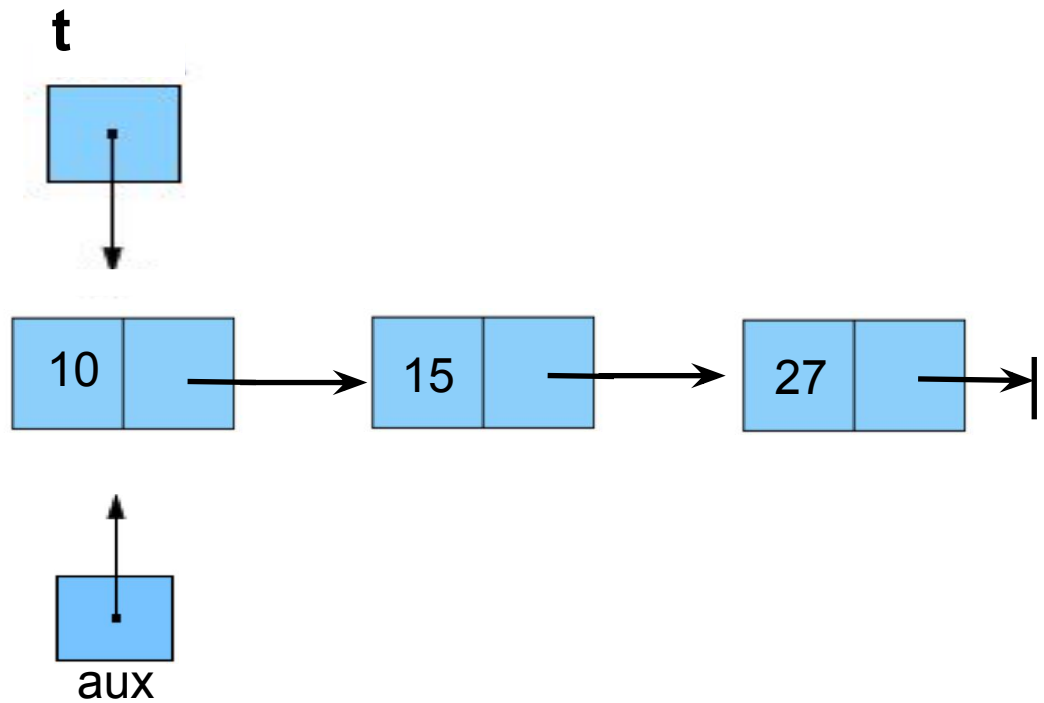


# OPERAÇÃO: empilhar

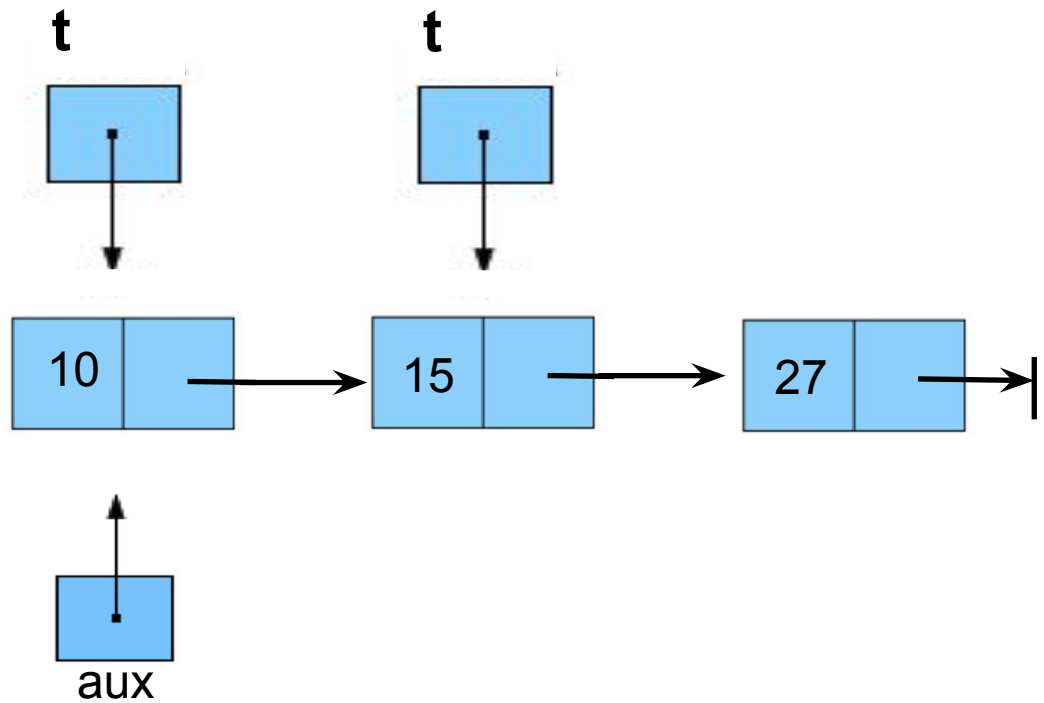
```
/*Função recebe o ponteiro para o topo da pilha, cria um novo nó  
(célula) para armazenar x e coloca esse nó como topo da pilha*/  
void empilhar(int x, celula *&lst)  
{  
    celula *nova;  
  
    nova = (celula *) malloc(sizeof (celula));  
    nova->chave = x;  
    nova->prox = lst;  
    lst = nova;  
}
```



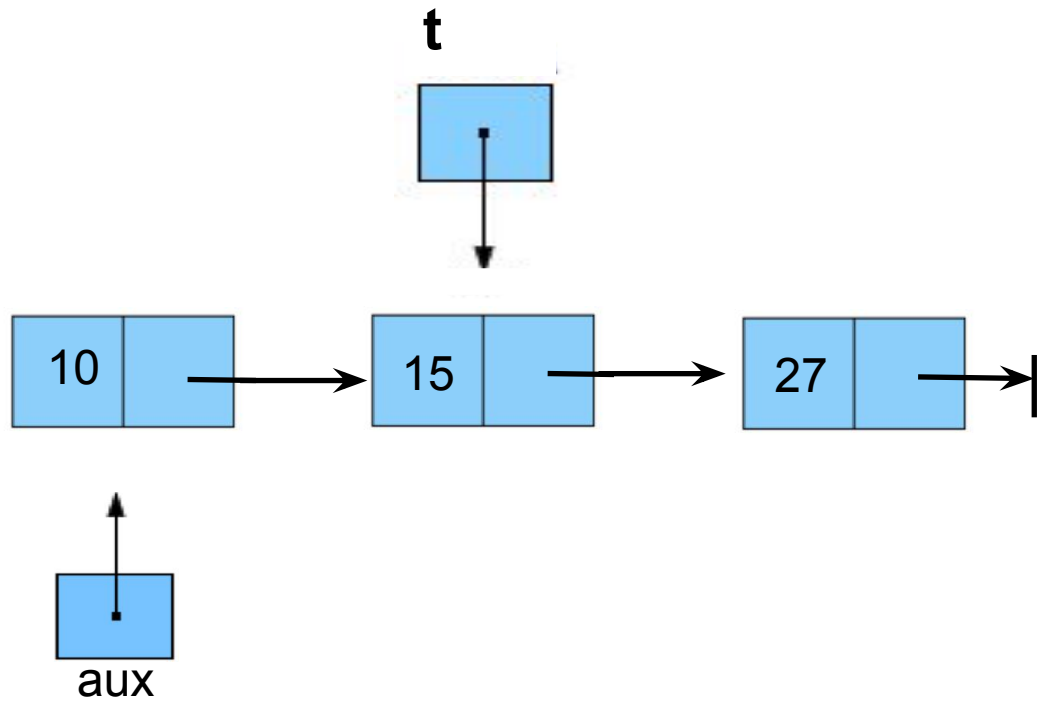
# OPERAÇÃO: desempilhar



# OPERAÇÃO: desempilhar



# OPERAÇÃO: desempilhar



# OPERAÇÃO: desempilhar

```
/*Função recebe o ponteiro para o topo da pilha, desempilha e retorna  
o valor do topo*/  
int desempilhar(celula *&lst)  
{  
    celula *aux;  
    int num;  
  
    if(lst == NULL)  
        return 0;  
    else{  
        num = lst->chave;  
        aux = lst;  
        lst = lst->prox;  
        free(aux);  
        return num;  
    }  
}
```