

Estrutura de Dados – SPOEDD1

Listas, Filas e Pilhas

Prof. Antonio Ferreira Viana

Regras

- 1) Inserção de elementos no final da estrutura (1 por 1)
- 2) Remoção a partir do início da estrutura (1 por 1)
- 3) Em casos especiais, é possível inserir elemento em outras posições na estrutura, exemplo: inserção ordenada (1 por 1)
- 4) Em casos especiais, é possível remover elemento de outra posição que não o início da estrutura (1 por 1)

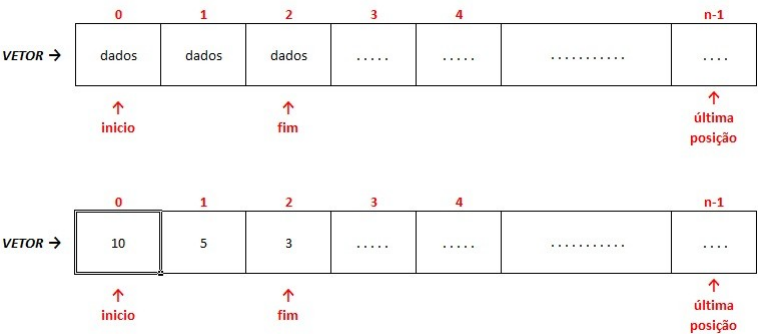
Regras e suas estruturas de dados

Estrutura de dados	Regras
Fila	1 e 2
Pilha	1 e 4 ou 2 e 3
Lista encadeada simples	1, 2, 3 e 4

3

Encadeamento simples

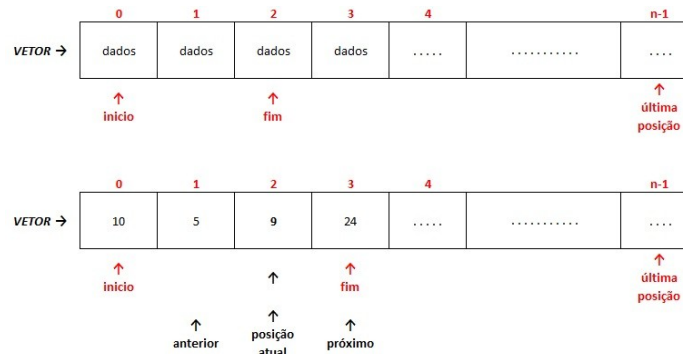
- Os elementos da lista possuem apenas uma ligação que liga o elemento atual com o elemento posterior (sucessor), pelo índice do vetor:



4

Duplamente encadeadas

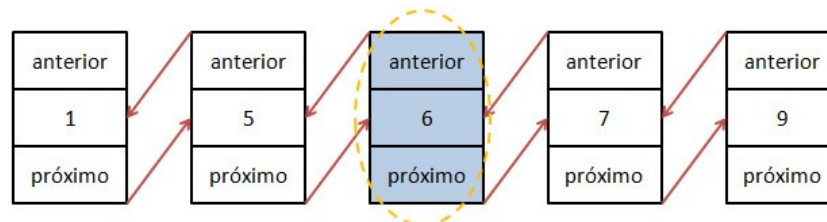
- Cada elemento possui um índice que aponta para o seu predecessor (anterior) e outro para o seu sucessor (próximo):



5

Ordenadas

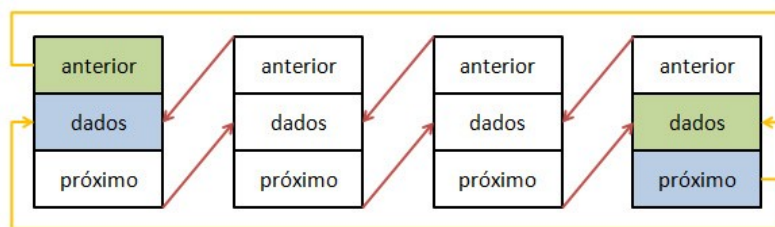
- A ordem linear da lista corresponde à ordem linear dos elementos, isto é, quando um novo elemento é inserido na lista ele deve ser colocado em tal posição que garanta que a ordem da lista será mantida; essa ordem pode ser definida por um campo da área de dados. Uma lista ordenada pode ser de encadeamento simples ou duplo, mas o princípio para ordenação é o mesmo:



6

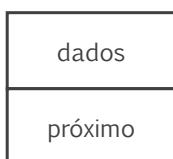
Circulares

- O ponteiro próximo do último elemento aponta para o primeiro; e o ponteiro anterior do primeiro elemento aponta para o último:



7

Estrutura dos elementos com ponteiro

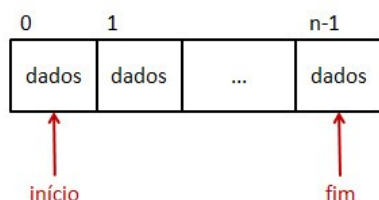


```
struct celula {
    int dado;
    struct celula *prox;
};
```

```
struct celula *fila;
```

8

Estrutura dos elementos com vetor



```
#define TAM_MAX 100
```

```
int fila[TAM_MAX];
```

```
int inicio, fim;
```

```
int i, j;
```

```
inicio = -1;
```

```
fim = -1;
```

} Indica que a estrutura
está vazia

9

Funcionalidades

Função/Procedimento para:

- Inserir elemento
- Remover elemento
- Exibir conteúdo
- Exibir primeiro/último elemento
- Exibir topo
- Inserir elemento de modo ordenado
- Remover elemento de posição específica

10

TAREFA – Individual para entregar até 06/10

- Implementar em linguagem C, utilizando vetores, as estruturas FILA e PILHA
- Funcionalidades a serem implementadas na estrutura FILA
 - Inserção, remoção, exibição dos elementos e suas respectivas posições na fila, verificação se a fila está vazia, verificação se a fila está cheia, localização de um elemento, retornando -1 se não estiver na fila e, a posição do elemento caso esteja na FILA.
- Funcionalidades a serem implementadas na estrutura PILHA
 - Inserção, remoção, Topo (mostra o elemento que está no topo da pilha mas não o remove) , verificação se a pilha está vazia, verificação se a pilha está cheia

11

Estrutura dos elementos com ponteiro

dados
próximo

```

struct celula {
    int dado;
    struct celula *prox;
};

struct celula * fila;
  
```

12

Encadeamento simples

- Os elementos da lista possuem apenas uma ligação que liga o elemento atual com o elemento posterior (sucessor), através de um ponteiro para o próximo elemento:
- Dada uma estrutura com um dado inteiro e um ponteiro para a estrutura, temos a declaração de uma lista como:

```
struct celula * fila;  
struct celula * lista;  
struct celula * pilha;
```

13

Encadeamento simples

- Alocação de memória:
- Uso de bibliotecas para alocação dinâmica
- As funções de alocação, retornam uma quantidade de memória em bytes ou retornam NULL em caso de erro de alocação
- Exemplo usando malloc:

```
struct celula * temp;  
  
temp = (struct celula *) malloc (sizeof(struct celula));
```

14

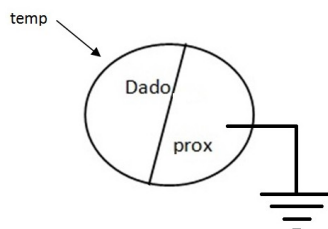
Representação gráfica

▪ Exemplo usando malloc:

```
struct celula * temp;
```

temp

```
temp = (struct celula *) malloc (sizeof(struct celula));
```



15

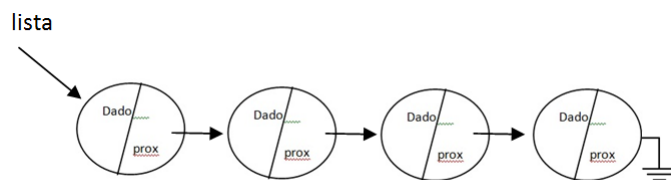
Representação gráfica

▪ Exemplo de lista montada:

```
struct celula * temp;
```

temp

```
struct celula * lista;
```



16

Representação gráfica

- Faça o passo a passo da inclusão e remoção de elementos em uma fila, que deverá conter os seguintes dados: 20, 10, 5, 30, 99, 7

Criação da fila:

temp



```
temp = (struct celula *)malloc( sizeof(struct celula));
```

```
temp->prox = NULL;
```

17

Representação gráfica

- Faça o passo a passo da inclusão e remoção de elementos em uma fila, que deverá conter os seguintes dados: 20, 10, 5, 30, 99, 7

Criação da fila:

fila

```
//PARA O PRIMEIRO ELEMENTO
```



```
temp -> dado = 20; // ou scanf("%d", &temp->dado);
```

```
temp = NULL;
```

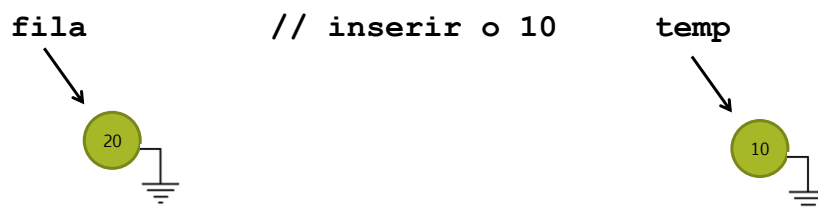
```
fila = temp;
```

18

Representação gráfica

- Faça o passo a passo da inclusão e remoção de elementos em uma fila, que deverá conter os seguintes dados: 20, 10, 5, 30, 99, 7

Criação da fila:

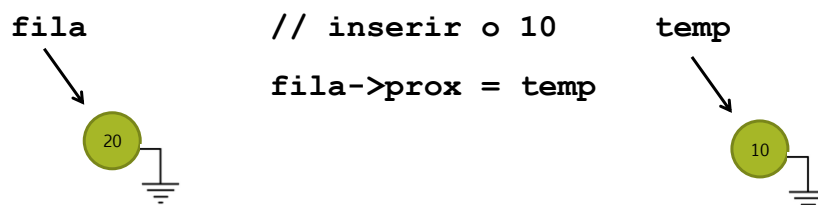


19

Representação gráfica

- Faça o passo a passo da inclusão e remoção de elementos em uma fila, que deverá conter os seguintes dados: 20, 10, 5, 30, 99, 7

Criação da fila:

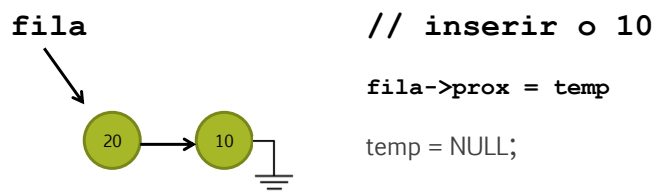


20

Representação gráfica

- Faça o passo a passo da inclusão e remoção de elementos em uma fila, que deverá conter os seguintes dados: 20, 10, 5, 30, 99, 7

Criação da fila:

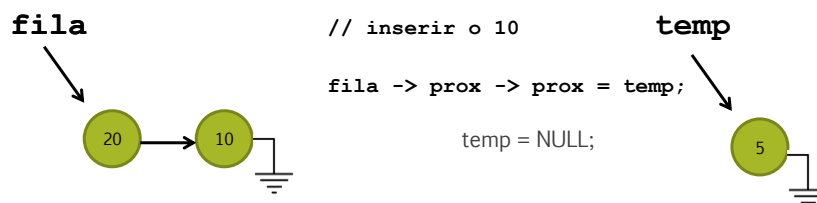


21

Representação gráfica

- Faça o passo a passo da inclusão e remoção de elementos em uma fila, que deverá conter os seguintes dados: 20, 10, 5, 30, 99, 7

Criação da fila:



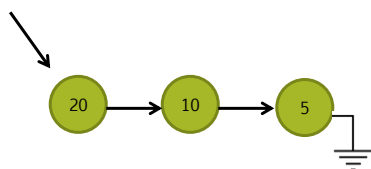
22

Representação gráfica

- Faça o passo a passo da inclusão e remoção de elementos em uma fila, que deverá conter os seguintes dados: 20, 10, 5, 30, 99, 7

Criação da fila:

fila



```
// inserir o 5
```

```
fila -> prox-> prox = temp;
```

```
temp = NULL;
```

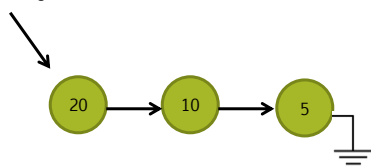
23

Representação gráfica

- Faça o passo a passo da inclusão e remoção de elementos em uma fila, que deverá conter os seguintes dados: 20, 10, 5, 30, 99, 7

Criação da fila:

fila



```
// inserir o 30, 99, 7...
```

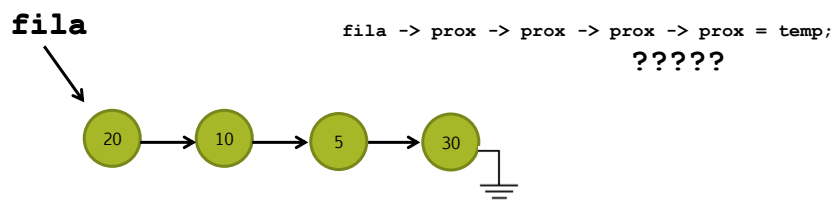
```
temp = NULL;
```

24

Representação gráfica

- Faça o passo a passo da inclusão e remoção de elementos em uma fila, que deverá conter os seguintes dados: 20, 10, 5, 30, 99, 7

Criação da fila:

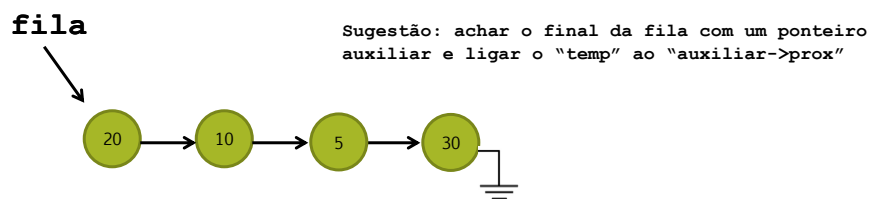


25

Representação gráfica

- Faça o passo a passo da inclusão e remoção de elementos em uma fila, que deverá conter os seguintes dados: 20, 10, 5, 30, 99, 7

Criação da fila:

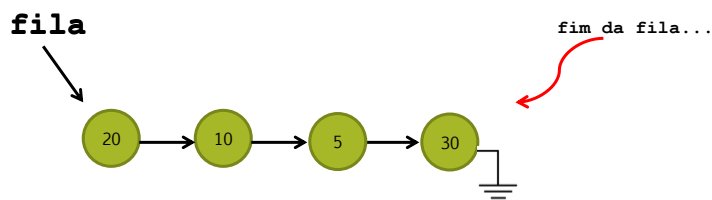


26

Representação gráfica

- Faça o passo a passo da inclusão e remoção de elementos em uma fila, que deverá conter os seguintes dados: 20, 10, 5, 30, 99, 7

Criação da fila:

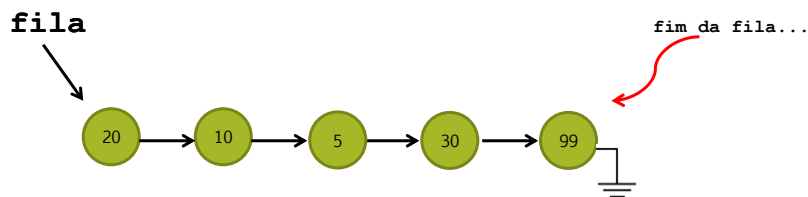


27

Representação gráfica

- Faça o passo a passo da inclusão e remoção de elementos em uma fila, que deverá conter os seguintes dados: 20, 10, 5, 30, 99, 7

Criação da fila:

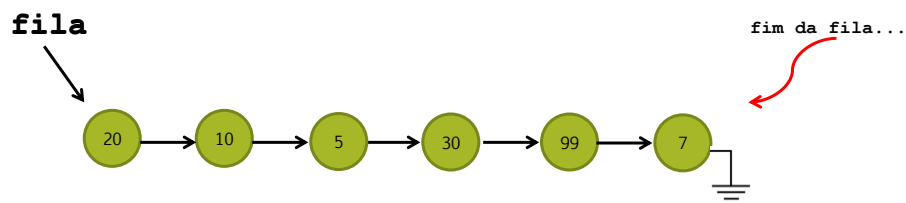


28

Representação gráfica

- Faça o passo a passo da inclusão e remoção de elementos em uma fila, que deverá conter os seguintes dados: 20, 10, 5, 30, 99, 7

Criação da fila:

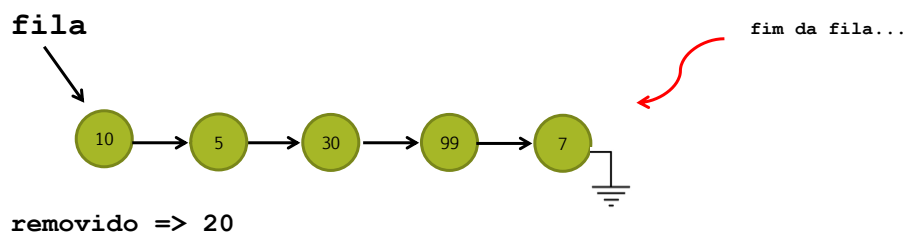


29

Representação gráfica

- Faça o passo a passo da inclusão e remoção de elementos em uma fila, que deverá conter os seguintes dados: 20, 10, 5, 30, 99, 7

Remoção de dados:



30

Representação gráfica

- **Faça o passo a passo da inclusão e remoção de elementos em uma fila, que deverá conter os seguintes dados: 20, 10, 5, 30, 99, 7**

A remoção neste exemplo ocorre até que a fique vazia, sendo que a sequência de remoções equivale a: 20, depois 10, depois 5, e assim por diante.

A representação do passo a passo graficamente também é chamada de Diagrama de Estados da estrutura.

31

Representação gráfica

- **Exercício em sala (com ponteiros):**
- **Faça o passo a passo da inclusão e remoção de elementos em lista simplesmente encadeada, lista ordenada e em uma pilha, sinalizando onde é o início, fim, e se for o caso, o topo da estrutura A sequência de inserções deve ser:**

20, 10, 5, 30, 99, 7

32

Implementação em C

- Como de costume, você deve implementar as estruturas lista, fila e pilha utilizando alocação dinâmica, em linguagem C.
- Não é necessário entregar esta implementação, mas é importante que seja realizada, pois será a base da implementação dos algoritmos de manipulação de árvores binárias.