

SIN 211 - Algoritmos e Estruturas de Dados (Filas)

Prof^o: Joelson Antônio dos Santos

Universidade Federal de Viçosa
Instituto de Ciências Exatas e Tecnológicas
Campus de Rio Paranaíba - MG

joelsonn.santos@gmail.com
Sala: BBT 233

3 de maio de 2018

Aula de Hoje

1 Filas

- Estática
- Dinâmica
- Fila de Prioridade

Créditos

O material desta aula é composto por adaptações e extensões dos originais gentilmente cedidos pelos professores **Moacir Pereira Ponti** e **Rachel Reis**.

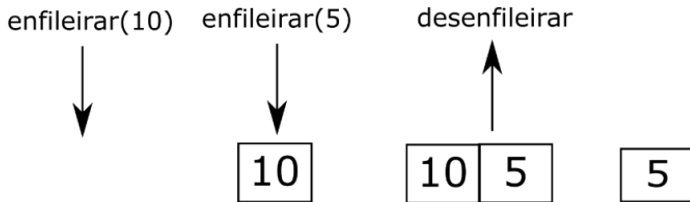
Filas



- Uma linha de espera que cresce somando elementos ao seu final e que diminui tomando elementos de sua frente.
- Ambas as extremidades são usadas.
- Uma estrutura do tipo FIFO (*first in/ first out*).

- Operações comuns.
 - Limpar fila.
 - Verificar se a fila está vazia.
 - Verificar se a fila está cheia (estática).
 - Enfileirar (*enqueue*).
 - Desenfileirar (*dequeue*).
 - Retornar primeiro elemento sem desenfileirar.

Fila Estática



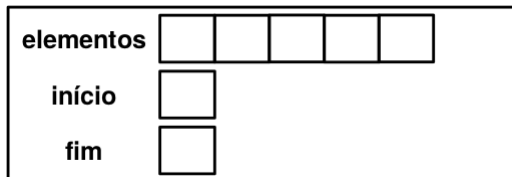
- Implementações:
 - **Estática:** É utilizado um vetor e duas variáveis “início” e “fim” para armazenar o índice do primeiro e do último elemento.
 - **Dinâmica:** É utilizado uma estrutura para definirmos os nós (células). Cada nó possui a informação e um ponteiro.

Fila Estática

- Implementada por meio de um vetor.
- Não é a melhor escolha para implementação!
- Enfileirar faz a parte de trás da fila se expandir.
- Desenfileirar faz a parte da frente se contrair.

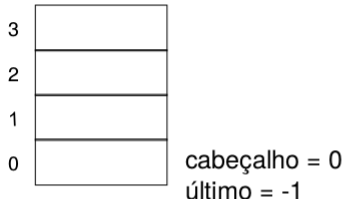
Fila Estática

- Como seria implementada uma fila estática?
- Quais valores seriam colocados para as variáveis de cabeçalho (*início*) e último (*fim*)?

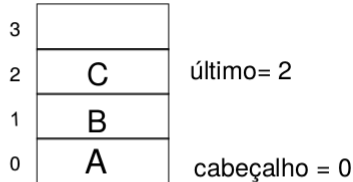


Fila Estática

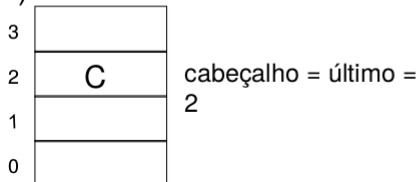
1) Fila vazia



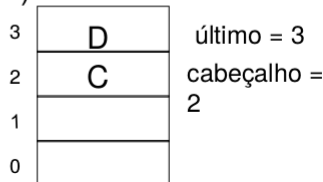
2) Insere A, B e C



3) Elimina dois itens

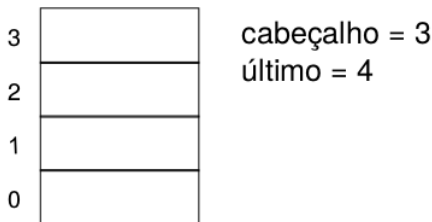


4) Insere novo item



Fila Estática

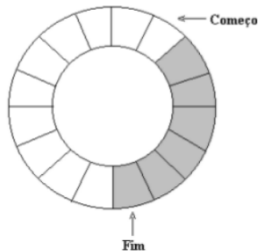
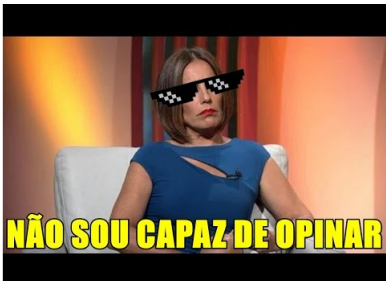
- Podemos chegar a uma situação onde a fila possui espaço mas não podemos adicionar elementos.



- Como resolver este problema?

Fila Estática

- Devemos visualizar uma fila estática como um **vetor circular**!
- É necessário que as operações de enfileirar e desenfileirar considerem a possibilidade de “fechar vetor”, ou seja, devem voltar ao início do vetor caso haja posições “livres”.

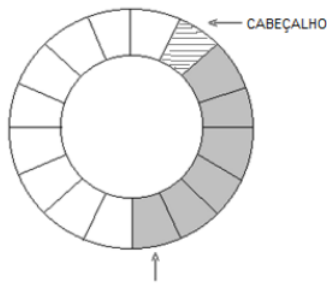


Fila Estática - Implementação

- Quando minha fila está vazia?
- Solução:
 - Abrir mão de um espaço na fila, criando um cabeçalho, que não possui um elemento válido. O primeiro elemento real da nossa fila é o elemento que sucede o cabeçalho!

Fila Estática - Implementação

- Se o cabeçalho for igual ao último, minha fila está **vazia!**
- Se o próximo índice para adicionar o elemento (último) for igual ao cabeçalho, minha fila está **cheia!**



Fila Estática - Implementação

```
#include <stdio.h>
#include <stdlib.h>
#define TAM 10

// definição
typedef struct fila{
    int elem[TAM];
    int inicio;
    int fim;
}FILA;

// inicializar fila estática
void inicializar(FILA* fila){
    fila->fim = 0;
    fila->inicio = fila->fim;
}
```


Fila Estática - Implementação

```
int filaVazia(FILA* fila){  
    return (fila->fim == fila->inicio);  
}  
  
int filaCheia(FILA* fila){  
    int prox = (fila->fim+1)%TAM;  
    return (prox == fila->inicio);  
}
```

Fila Estática - Implementação

- Enfileirar

```
int enfileirar(FILA* fila, int elemento){
    int prox;
    if(filaCheia(fila)){
        printf("\n Fila Cheia");
        return 0;
    }
    prox = (fila->fim + 1)%TAM;
    fila->elem[prox] = elemento;
    fila->fim = prox;
    return 1;
}
```

Fila Estática - Implementação

- Desenfileirar

```
int desenfileirar(FILA* fila){
    int removido = -1;
    if(filaVazia(fila)){
        printf("\n Fila Vazia");
        return removido;
    }
    fila->inicio = (fila->inicio + 1)%TAM;
    removido = fila->elem[fila->inicio];
    return removido;
}
```

Fila Estática - Implementação

- O que seria necessário fazer para imprimir todos os elementos de uma fila?



Fila Estática - Implementação

- **Exercício:** Crie duas funções de imprimir os elementos de uma fila de inteiros da seguinte maneira:
 - Desenfileirando toda a fila.
 - Usando uma fila auxiliar e retornando os elementos para a fila original ao final da impressão.

Fila Estática - Aplicações

- Tanto enfileirar quanto desenfileirar podem ser executadas em tempo constante $O(1)$.
- Filas são frequentemente usadas em simulações, uma vez que existe uma **teoria das filas** matematicamente sofisticada, onde vários cenários são analisados e modelos que usam filas são construídos.

Fila Estática - Exemplo

- Peças em uma linha de montagem no processo de criação de uma máquina, caminhões esperando por atendimento em uma estação de pesagem, armazéns, correios, bancos, etc.

Fila Dinâmica

- A implementação dinâmica é baseada na implementação de uma lista simplesmente encadeada.
- A diferença é que a fila precisa ter acesso às duas extremidades para evitar a necessidade de percorrê-la quando formos enfileirar ou desenfileirar.

Fila Dinâmica

- Definindo uma fila dinâmica:

```
typedef struct sCell{  
    int info;  
    struct sCell * prox;  
}CELULA;  
  
typedef struct fila{  
    CELULA* inicio;  
    CELULA* fim;  
}FILA;
```

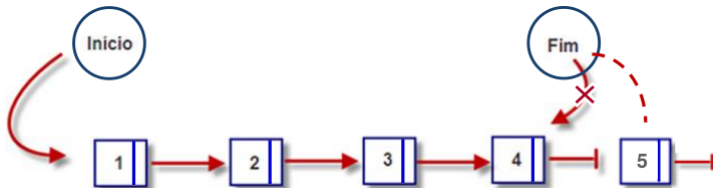
- Na função principal, criamos nossa fila. Ela deve ser de que tipo?

Fila Dinâmica

- Inicializar:
 - Responsável por inicializar os ponteiros “início” e “fim” de uma fila com o valor **NULL**.
- Verifica se está vazia:
 - Responsável por retornar se a fila está vazia (não possui elementos). Retorna 1 se estiver vazia e 0 caso contrário.

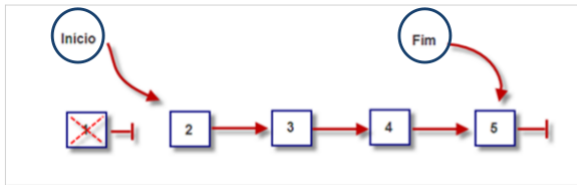
Fila Dinâmica - Operações

- Enfileirar:
 - Responsável por criar, inicializar e inserir um elemento no final da fila.
- Quais os parâmetros necessários para a criação desta função?



Fila Dinâmica - Operações

- Desenfileirar:
 - Responsável por armazenar o dado que será removido, atualizar o primeiro elemento da fila, liberar a memória do “antigo” primeiro elemento e retornar a informação do elemento removido.
- Qual o tipo de retorno desta função?



Fila com Prioridade

- Em muitas situações, filas simples são inadequadas.
- Pode existir uma necessidade de validação segundo algum critério de prioridade.
- O elemento com a maior prioridade deve ser o primeiro a ser removido da fila.



Fila com Prioridade

- O problema com uma fila com prioridades é encontrar uma implementação eficiente que permita:
 - Rapidez na inserção.
 - Rapidez na remoção.
- Duas variações (simples) de filas encadeadas:
 - Elementos enfileirados de forma ordenada.
 - Enfileirar normalmente um elemento (fila fica desordenada baseado na prioridade).

Fila com Prioridade

- Fila ordenada:
 - Enfileirar $O(n)$.
 - Desenfileirar $O(1)$.
- Fila desordenada:
 - Enfileirar $O(1)$.
 - Desenfileirar $O(n)$.

Bibliografia Básica

- DROZDEK, Adam. Estrutura de Dados e Algoritmos em C++. Editora Pioneira Thomson Learning, 2005.
 - Pág 130 (Fila)
 - Pág 138 (Filas com Prioridades)
- TENENBAUM A., LANGSAM Y. e AUGENSTEIN M. J. Estrutura de Dados usando C. Editora Makron, 1995.
 - Pág 209 (Fila)