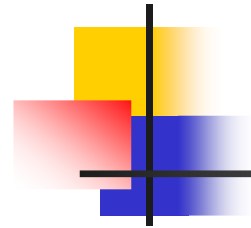


Estruturas de Dados – EDDA3

Fila Dinâmica

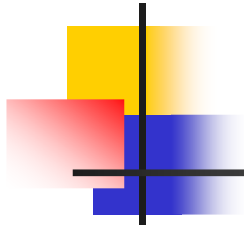
Prof. Marcelo Zorzan

Profa. Melissa Zanatta



Aula de Hoje

- TAD Fila



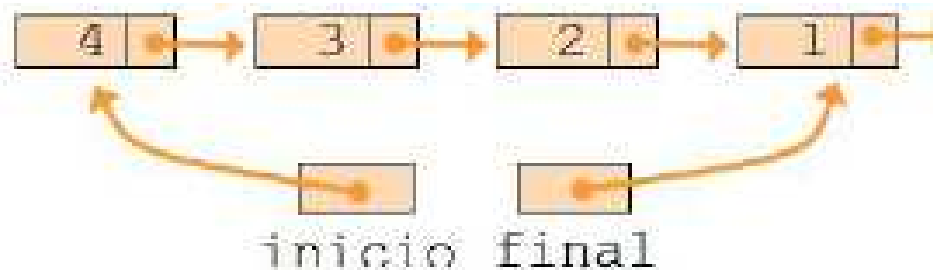
TAD Fila

- Sequência de objetos, todos do mesmo tipo , sujeito às seguintes regras de comportamento:
 - 1) Sempre que solicitamos a remoção de um elemento, o elemento removido é o primeiro da sequência
 - 2) Sempre que solicitamos a inserção de um novo objeto, o objeto é inserido no fim da sequência.
- Em resumo...

“O elemento removido é sempre o que está lá há mais tempo”.

Implementação – lista encadeada

- Para implementar uma fila como uma lista encadeada, a fila precisa ter acesso às suas duas extremidades
 - Para isso será necessário armazenar dois ponteiros





Implementação – lista encadeada

- Definição do tipo Fila

```
typedef struct cell {  
    int info;  
    struct cell *prox;  
}CELULA;
```

```
main (){  
    CELULA *inicio;  
    CELULA *fim;  
    ...  
}
```

Outra abordagem:

```
typedef struct sFila {  
    CELULA* inicio;  
    CELULA* fim;  
}Fila;
```



Implementação – lista encadeada

- Inicializa
 - Nome da função: inicializa
 - Tipo de retorno: void
 - Descrição: Inicializa o ponteiro “inicio” e “fim” com valor NULL
- Verifica se a lista está vazia
 - Nome da função: vazia
 - Tipo de retorno: int
 - Retornar 1 se a fila estiver vazia; 0 caso contrário.



Implementação – lista encadeada

■ Enfileirar

- Nome da função: enfileirar
- Tipo de retorno: void
- Número de parâmetros: 2
 - 1º - Variável do tipo "Fila*"
 - 2º - Variável do tipo: <tipo_da_informação>
- Descrição: função responsável por criar, inicializar, inserir um elemento no final da fila
- Levar em consideração os seguintes casos:
 - 1) Não foi possível alocar memória para o novo elemento
 - 2) A fila está vazia
 - 3) A fila não está vazia



Implementação – lista encadeada

- Desenfileirar
 - Nome da função: desenfileirar
 - Tipo de retorno: <tipo_da_informação>
 - Parâmetros:
 - 1) Variável do tipo "Fila*"
 - Descrição: função responsável por remover um elemento no início da fila.
 - Considerar as seguintes situações:
 - A fila está vazia (exibir mensagem de erro)
 - Depois de remover o elemento a fila ficou vazia
 - Depois de remover o elemento a fila não ficou vazia



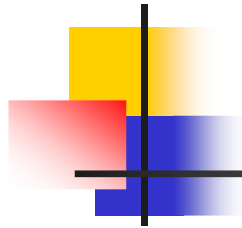
Filas com prioridades

- Situações em que filas simples são inadequadas:
 - Correio: uma pessoa com incapacidade física pode ter prioridades sobre outras;
 - Cabines de pedágio: veículos como “carros de polícia”, “ambulância”, “carros de bombeiro” podem passar imediatamente, mesmo sem pagamento.
- Em filas com prioridades, os elementos são inseridos/removidos da fila de acordo com suas prioridades.



Filas com prioridades

- Problema: encontrar uma implementação eficiente que permita uma rápida colocação e uma rápida retirada da fila.
 - Correio: uma pessoa com incapacidade física pode ter prioridades sobre outras
 - Cabines de pedágio: veículos como “carros de polícia”, “ambulância”, “carros de bombeiro” podem passar imediatamente, mesmo sem pagamento.
- Em filas com prioridades, os elementos são retirados da fila de acordo com suas prioridades e suas posições correntes na fila.



Leituras Recomendadas

- 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)

- FEOFILOFF, Paulo. Algoritmos em Linguagem C. Editora Campus, 2009.

→ Pág 31 (Fila)