ESTRUTURAS DE DADOS - FUNDAMENTOS

Parte II

Filas (queue)

Fila é uma lista linear restrita pelo fato das operações de acesso e remoção serem realizadas em uma das extremidades chamada início e a operação de inserção ser realizada na outra extremidade chama fim.

Propriedade: o primeiro elemento inserido será o primeiro a ser removido. (FIFO-first-in, first-out).

Ìmplementação

- Alocação Seqüencial
 - Utilização de arranjos
- Alocação Ligada
 - Utilização de apontadores

Implementação em um vetor - alocação següencial:

Suponha que uma fila está armazenada em um arranjo elementos[0..MAX-1]. Vamos supor que os elementos da pilha são inteiros. A parte do vetor ocupada pela fila será:

Elementos para um arranjo de MAX=10

Valores			8	3	1	5	7			
Índices	0	1	2	3	4	5	6	7	8	9
			Ι				F			

Inserção:

Note que índice F indica o Fim da.

Todas as inserções são realizadas na posição indicada por F. Assim para a próxima inserção devemos incrementar a posição F (F=F+1).

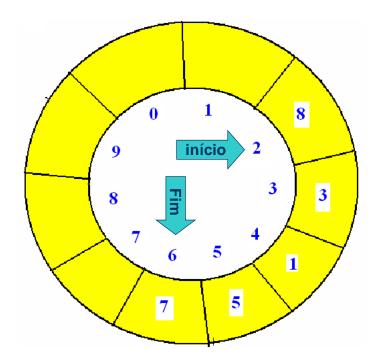
Remoção:

Note que índice I indica inicio da fila.

O primeiro elemento a ser removido está no início da fila, indicada por I. Logo, para remover o elemento basta incrementar o valor de I (I = I + 1)

Representação Circular

Observe que os índices F (fim) e I (inicio) estão sendo incrementados em uma unidade cujos valores atingem MAX. Assim, ao atingirem MAX, eles passam a assumir o valor 0 (zero), denotando a utilização de um arranjo circular, conforme mostra a figura, abaixo.



Fila em Alocação Sequencial (Arranjo Circular

```
Implementação em C/C++
#ifndef FILA_H_INCLUDED
#define FILA_H_INCLUDED
template <typename Tipo>
struct Fila{
 private:
  Tipo *v;
  int tamanho;
  int qtde;
  int inic;
  int fim;
 public:
  Fila(int tam){
   tamanho = tam;
    v = new Tipo[tamanho];
    qtde=0;
    fim=-1;
    inic=0;
  ~Fila(){
    delete v;
  void insere(Tipo x){
    fim++;
    if(fim==tamanho){
       fim=0;
    v[fim]=x;
    qtde++;
  Tipo remover(){
    Tipo temp=v[inic];
    inic++;
    if(inic==tamanho){
       inic=0;
```

```
qtde--;
    return temp;
  Tipo primeiro(){
    return v[inic];
  bool filaCheia(){
    return qtde==tamanho;
  bool filaVazia(){
    return qtde==0;
  int getTamanho(){
    return tamanho;
  int getQtde(){
    return qtde;
  int getInic(){
    return inic;
  int getFim(){
    return fim;
  Tipo getValor(int pos){
    return v[pos];
#endif // FILA_H_INCLUDED
```

Exercício 1:

Considere os protótipos, definidos abaixo, para uma estrutura de dados do tipo fila:

Procedimento Insere(X:inteiro);

Função Remove:inteiro;

Ex:Mostre a situação de uma fila, após cada uma das seguintes operações:

- 1. insere(5)
- 2. insere(9)
- 3. insere(2)
- 4. remove()
- 5. insere(remove())
- 6. remove()
- 7. insere(1)
- 8. insere(3)
- 9. remove()
- 10. remove();
- 11. insere(7)
- 12. insere(remove())

Exercício 2: Trabalho Prático

Mostre uma aplicação que simule as operações de uma fila, implementada em alocação sequencial com representação circular. Esta simulação, executada um número n de vezes deve temporizar aleatoriamente um tempo de espera para as operações insere e remove. Mostre a situação dessa fila após cada operação.