Estrutura de Dados

Prof. Adriano Teixeira de Souza

Filas

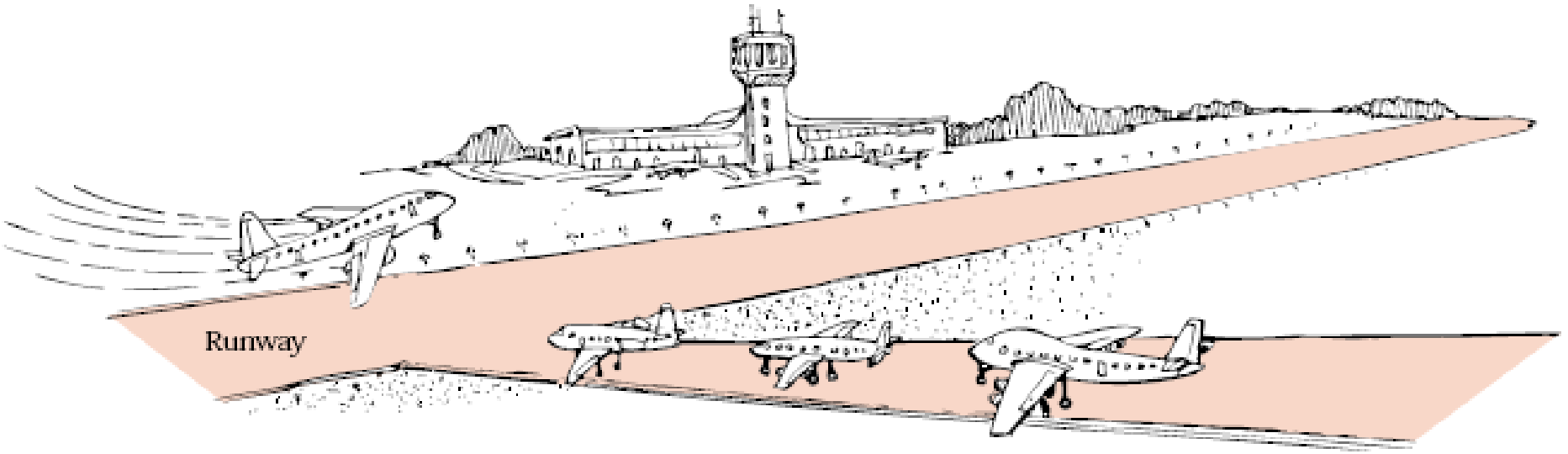
- São listas lineares que adotam a política FIFO (First In First Out – o primeiro que entra é o primeiro que sai) para a manipulação de elementos.
- As inserções são feitas no final da fila.
- As remoções são feitas no início da fila.
- A consulta na fila é feita desenfileirando elemento a elemento até encontrar o elemento desejado ou chegar ao final da fila.

Filas
:: Aplicações

- Alocação de recursos para impressão de documentos em uma impressora (spooler de impressão).
- Atendimento de processos requisitados ao um sistema operacional.
- Ordenação do encaminhamento dos pacotes em um roteador.
- Buffer para gravação de dados em mídia.

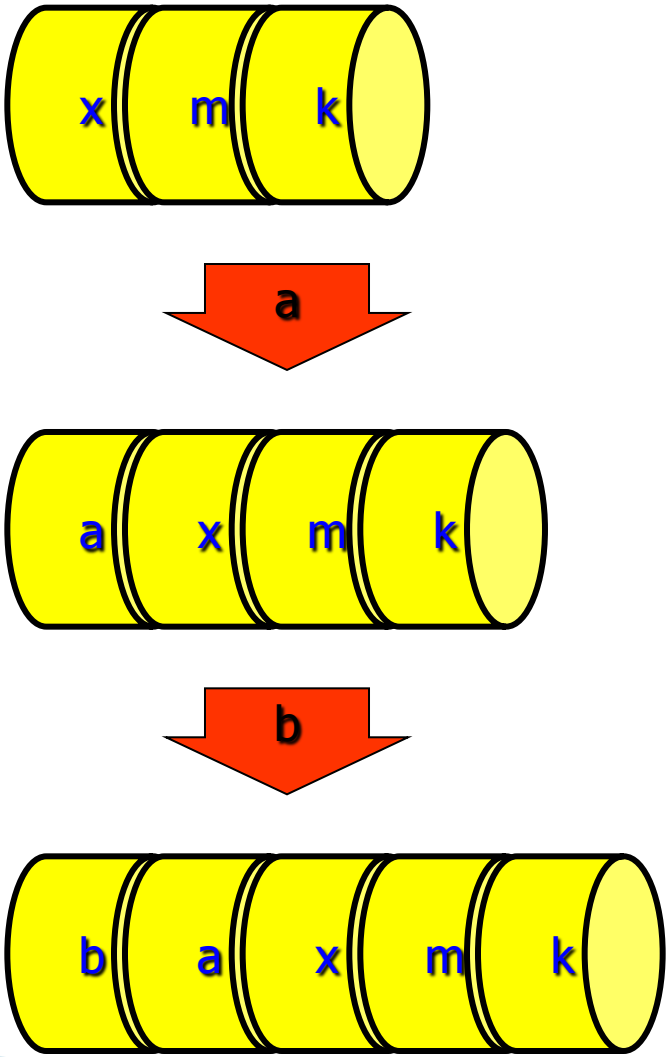
Filas
:: Aplicações

fila para pouso

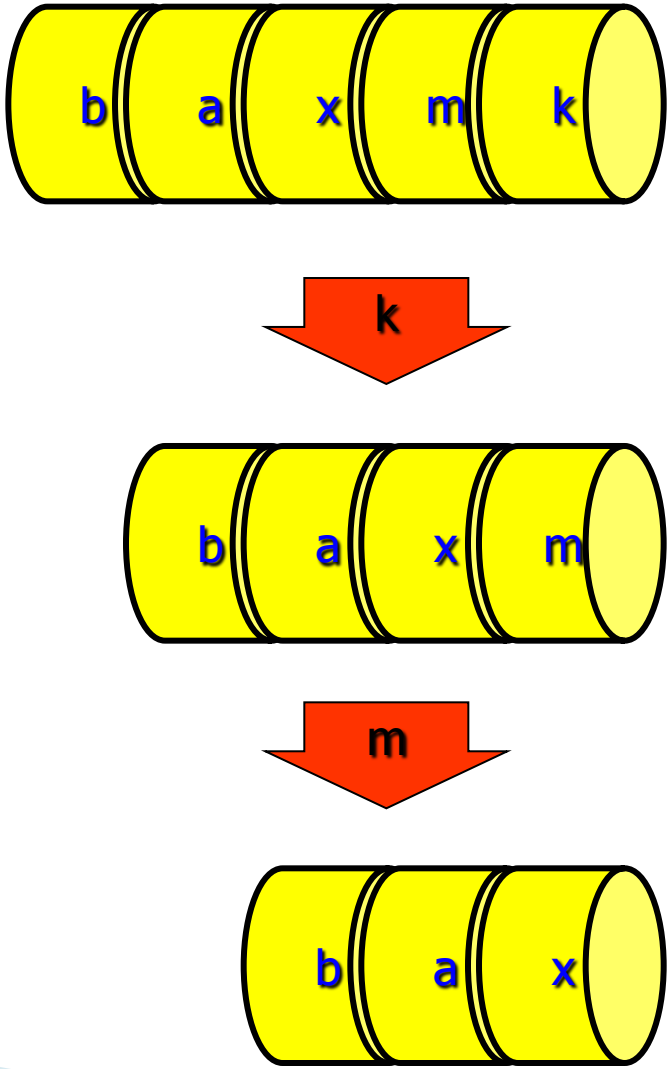


fila para decolagem

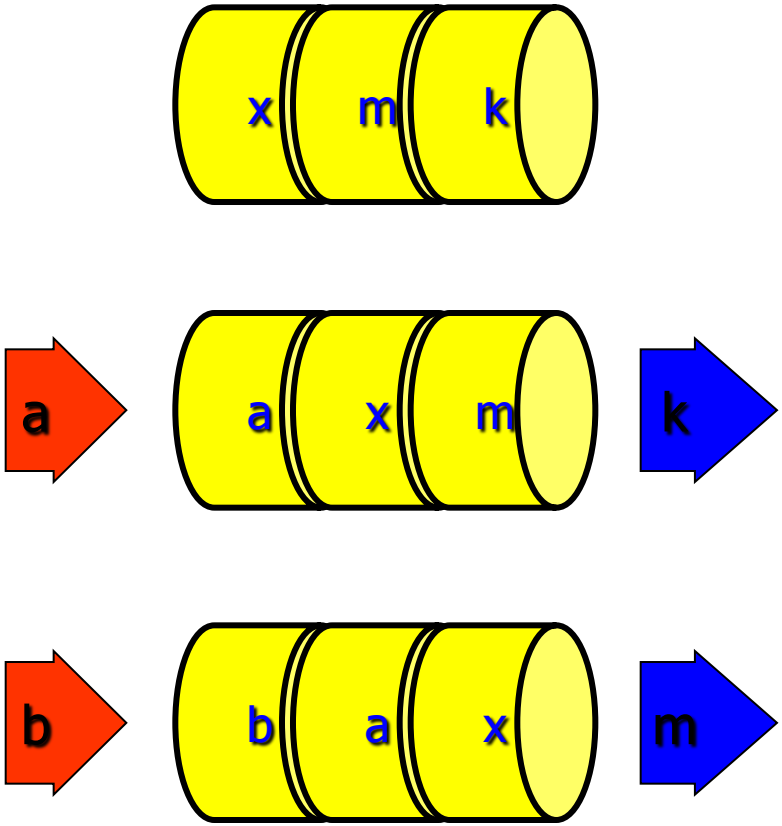
Filas de tamanho variável
:: Inserção



Filas de tamanho variável
:: Remoção



Filas de tamanho fixo



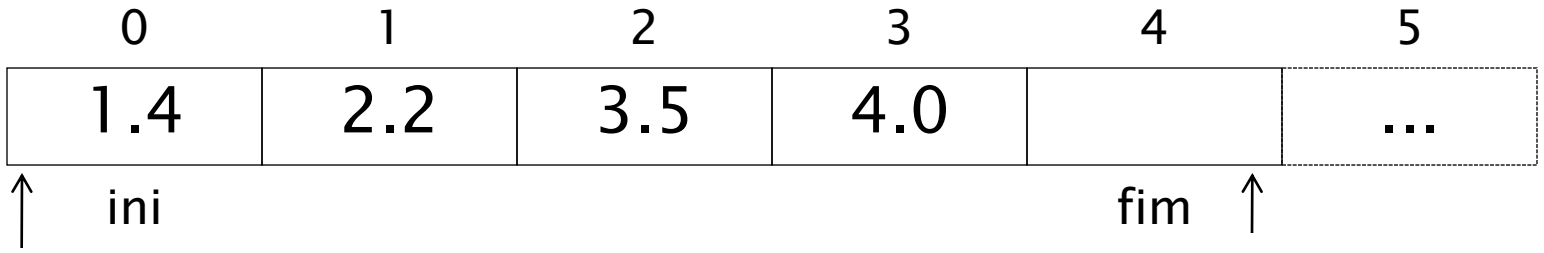
Filas
:: Operações básicas

- Criação
- Destruição
- Inserção de um elemento
- Remoção de um elemento
- Localização de um elemento para consulta ou alteração
- Ordenação de uma lista
- Intercalação de duas listas
- Concatenação de duas listas
- Divisão de uma lista em duas

Implementação de pilha com vetor

- Para isso devemos fixar o número máximo N de elementos na fila.
- Note que o processo de inserção e remoção em extremidades opostas fará com que a fila “ande” no vetor.
- Exemplo: se inserirmos os elementos 1.4, 2.2, 3.5, 4.0 e depois retirarmos dois elementos, a fila não estará mais nas posições iniciais do vetor.

Implementação de fila com vetor

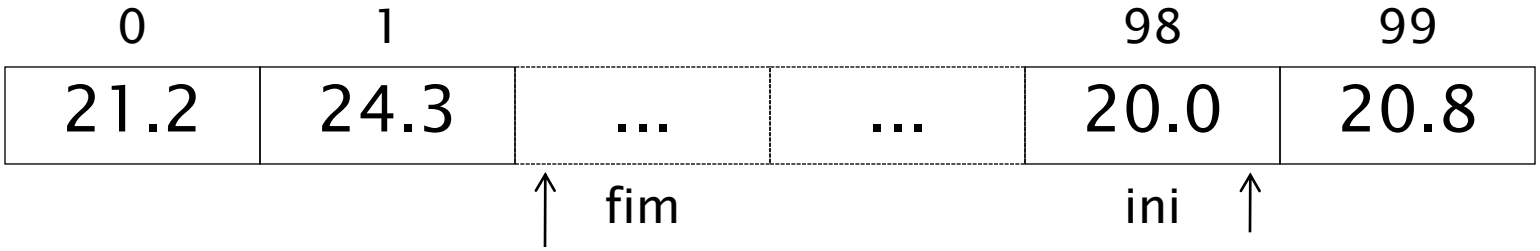


Implementação de fila com vetor

- Observamos que em dado instante, a parte ocupada do vetor pode chegar à última posição.
- Para reaproveitar as primeiras posições livres do vetor sem implementarmos uma re-arrumação trabalhosa dos elementos, podemos incrementar as posições do vetor de forma “circular”: se o último elemento ocupa a última posição do vetor, inserimos os novos elementos a partir do início do vetor.

Implementação de fila com vetor

- Desta forma, em um dado momento, poderíamos ter quatro elementos, 20.0, 20.8, 21.2 e 24.3, distribuídos dois no fim e dois no início do vetor.



Implementação de fila com vetor

- Para essa implementação, os índices do vetor são incrementados de maneira que seus valores progridam “circularmente”.
- Desta forma, se temos 100 posições no vetor, os valores dos índices assumem os seguintes valores:

- 0, 1, 2, 3, ..., 98, 99, 0, 1, 2, 3, ..., 98, 99, 0, 1, ...

Implementação de fila com vetor

- Função auxiliar responsável por incrementar o valor de um índice.
- Esta função recebe o valor do índice atual e fornece com valor de retorno o índice incrementado, usando o incremento circular.

```
int incr (int i) {
      if (i == N-1)
            return 0;
      else
            return i+1;
}
```

Ou, dispensando a função: i=(i+1)%N;

A estrutura fila

- Podemos declarar uma estrutura tipo fila como sendo uma estrutura com três componentes:
  - Um vetor **vet** de tamanho **N**,
  - Um índice **ini** para o início da fila
  - Um índice **fim** para o fim da fila
- Onde,
  - **ini** marca a posição do próximo elemento a ser retirado da fila;
  - **fim** marca a posição (vazia), onde será inserido o próximo elemento

- Desta forma a fila vazia se dá **ini == fim** e a fila cheia se caracteriza por ter **fim** e **ini** em posições consecutivas (circularmente): **incr(fim) == ini.**

A estrutura fila

- A estrutura fila pode ser dada por:

```
#define N 100

typedef struct {
  int ini, fim;
  float vet[N];
} Fila ;
```

- A função para criar:

```
Fila* cria (void)
{
  Fila* f = (Fila*)
  malloc(sizeof(Fila));
  f->ini = f->fim = 0; //inicializa
  return f;
}
```

Função Insere

- Para inserir um elemento na fila, usamos a próxima posição livre do vetor, indicada por **fim**. Devemos verificar se há espaço para a inserção de um novo elemento (utilizamos vetor)

```
void insere (Fila* f, float v)
{
        if (incr(f->fim) == f->ini){ // fila cheia
                printf("Capacidade da fila estourou");
                exit(1); //aborta o programa
        }
        // insere elemento na próxima posição livre
        f->vet[f->fim] = v;
        f->fim = incr(f->fim);
}
```

Função Retira

- A função para retirar o elemento no início da fila fornece o valor do elemento retirado como retorno. Verifica-se antes se a fila está ou não vazia:

```
float retira (Fila* f) {
       float v;
       if (vazia(f)) {
              printf("Fila vazia.\n");
              exit(1); // aborta programa
       }
       //retira elemento no inicio
       v = f->vet[f->ini];
       f->ini = incr(f->ini);
       return v;
}
```

Filas :: Funções Vazia e Libera

- A função que verifica se a pilha está vazia pode ser dada por:

```
int vazia (Fila* f)
{
    return (f->ini == f->fim);
}
```

- Finalmente, a função para liberar a memória alocada pela fila:

```
void libera (Fila* f)
{
    free (f);
}
```

Filas :: Função Imprime

- Para testar o código, pode ser útil implementarmos um função que imprima os valores armazenados na fila. A ordem de impressão adotada é do início para o fim.

```
// versão com vetor
void imprime (Fila* f){
   int i;
   for (i=f->ini; i!=f->fim; i=incr(i)){
      printf("%f\n", f->vet[i]);
    }
}
```

Filas :: Exemplo de utilização

```
main(){

    Fila* f = cria();
    insere (f,20.0);
    insere (f,20.8);
    insere (f,20.2);
    insere (f,20.3);
    imprime (f);
    printf ("Primeiro elemento: %f\n", retira(f));
    printf ("Segundo elemento: %f\n", retira(f));
    printf ("Configuracao da fila:\n");
    imprime (f);
    libera (f);

    system("pause");

}
```