Estrutura de Dados 1

Prof. Igor Calebe Zadi
igor.zadi@ifsp.edu.br

INSTITUTO FEDERAL
São Paulo
Campus Catanduva

Filas especiais

INSTITUTO FEDERAL
São Paulo
Campus Catanduva

Implementações especiais para Filas

- Fila com prioridades
- Filas circulares
- Deque

Filas com prioridades

- Cada elemento tem prioridade

- Ordem de atendimento depende da prioridade
- Estratégias:
  - Inserção ordenada, atendimento normal
  - Inserção normal, atendimento com busca

Filas com prioridades

- Cada elemento possui um campo para armazenar a prioridade:

```
typedef struct Noh {
    int dado;
    int prioridade;
    struct Noh* prox;
} Noh;
```

Filas com prioridades

- Estratégias de implementação:
  - Inserção ordenada:
    - Procura a posição do elemento na fila
  - Pode demorar mais para inserir um elemento na fila
  - O atendimento será "normal" - retirar o primeiro elemento da fila

Filas com prioridades - inserção

```
void inserirOrdenado(Noh** inicio, int dado, int prioridade) {
    Noh* novo = malloc(sizeof(Noh));
    novo->dado = dado;
    novo->prioridade = prioridade;
    novo->prox = NULL;
    if (*inicio == NULL || prioridade < (*inicio)->prioridade) {
        novo->prox = *inicio;
        *inicio = novo;
        return;
    }

    Noh* atual = *inicio;
    while (atual->prox && atual->prox->prioridade <= prioridade) {
        atual = atual->prox;
    }
    novo->prox = atual->prox;
    atual->prox = novo;
}
```

Filas com prioridades - remoção

```
int removerComBusca(Noh** inicio) {
    if (*inicio == NULL) return -1;
    Noh *atual = *inicio, *anterior = NULL;
    Noh *maior = atual, *antMaior = NULL;

    while (atual != NULL) {
        if (atual->prioridade < maior->prioridade) {
            maior = atual;
            antMaior = anterior;
        }
        anterior = atual;
        atual = atual->prox;
    }

    if (antMaior == NULL) {
        *inicio = maior->prox;
    } else {
        antMaior->prox = maior->prox;
    }
    int valor = maior->dado;
    free(maior);
    return valor;
}
```

INSTITUTO FEDERAL
São Paulo
Campus Catanduva

Fila circular

- Último nó aponta para o primeiro
- Usa ponteiro para o fim
- Aplicações:
  - Buffers circulares
  - Sistemas de tempo real

Fila circular

```
void enfileirar(Noh** fim, int dado) {
    Noh* novo = malloc(sizeof(Noh));
    novo->dado = dado;
    if (*fim == NULL) {
        novo->prox = novo;
        *fim = novo;
    } else {
        novo->prox = (*fim)->prox;
        (*fim)->prox = novo;
        *fim = novo;
    }
}
```

Fila circular

```
int desenfileirar(Noh** fim) {
    if (*fim == NULL) return -1;

    Noh* inicio = (*fim)->prox;
    int valor = inicio->dado;

    if (inicio == *fim) {
        free(inicio);
        *fim = NULL;
    } else {
        (*fim)->prox = inicio->prox;
        free(inicio);
    }
    return valor;
}
```

Exemplos

- Sistema Unesp

Deque

- Inserção/remoção nas duas pontas
- Lista duplamente encadeada
- Tipos:
  - Entrada restrita
  - Saída restrita
- Operações:
  - Inserir e remover no início
  - Inserir e remover no final

Implementação de uma fila com pilhas

- É possível implementar uma fila utilizando duas pilhas
- Uma controlará as entradas e outra as saídas
- enqueue → empilha
- dequeue → desempilha (ou transfere)

Implementação de uma fila com pilhas

```
void push(Noh** topo, int dado);
int pop(Noh** topo);

void enqueue(Noh** entrada, int dado) {
    push(entrada, dado);
}

int dequeue(Noh** entrada, Noh** saida) {
    if (*saida == NULL) {
        while (*entrada) {
            push(saida, pop(entrada));
        }
    }
    return pop(saida);
}
```

Exercício

- Simulações:
  1) Suponha uma **fila circular** inicialmente vazia, com capacidade ilimitada (não há limite de tamanho).
     - Considere as seguintes operações:
       - enfileirar(10); enfileirar(20); enfileirar(30); desenfileirar() enfileirar(40); enfileirar(50); desenfileirar(); desenfileirar()

  2) Suponha uma fila com prioridades
     - Considere as operações:
       - inserir(5, prioridade=3); inserir(8, prioridade=1); inserir(4, prioridade=2); remover(); inserir(2, prioridade=0); remover(); remover()

  3) Realize as operações do exercício 1 para uma fila que utiliza a implementação de duas pilhas.

Apresente os estados das filas a cada operação

INSTITUTO FEDERAL
São Paulo
Campus Catanduva

Observações sobre o material eletrônico

- O material ficará disponível na pasta compartilhada que é acessada sob convite
- O material foi elaborado a partir de diversas fontes (livros, internet, colegas, alunos etc.)
- Alguns trechos podem ter sido inteiramente transcritos a partir dessas fontes
- Outros trechos são de autoria própria
- Esta observação deve estar presente em qualquer utilização do material fora do ambiente de aulas do IFSP - Catanduva

INSTITUTO FEDERAL
São Paulo
Campus Catanduva