

Respostas Lista de Exercícios 02 sobre Filas

Sistemas de Informação - UFG
Aluno: Marcos Vinicius Vieira da Silva
Matrícula: 147255

A maioria dos exercícios foram criados utilizando a IDE CLion e para cada um foi criado um projeto (pasta) com os arquivos .c que será enviada junto com esse PDF.

1 - R: pasta: exercício-1

2 - R: pasta: exercício-2

3 - R: pasta: exercício-3

4 - R:

Para que uma sequência de operações mantenha uma fila vazia sem provocar underflow:

- O número de operações **Enfileira** deve ser igual ao número de operações **Desenfileira**.

Para que a sequência deixe a fila não vazia inalterada:

- O número de operações **Enfileira** deve ser maior que o número de operações **Desenfileira**.

5 - R:

Vantagens:

Reduz a quantidade de deslocamentos, pois só desloca os elementos para o início do vetor quando o vetor está cheio. Melhora o desempenho das operações de Desenfileira, pois não há necessidade de mover elementos a cada remoção.

Desvantagens:

O deslocamento total de todos os elementos da fila pode ser custoso em termos de tempo quando o vetor atinge a capacidade máxima. Ainda possui a limitação de espaço de um vetor fixo, o que pode ser inadequado para filas com muitos elementos.

```
C/C++
// Função para verificar se a fila está vazia
bool filaVazia(Fila *f) {
    return f->inicio == f->fim;
}

// Função para enfileirar um elemento (com deslocamento somente quando o vetor está cheio)
bool enfileira(Fila *f, TipoItem item) {
    if (f->fim == MAX) {
        // Desloca os elementos para o início do vetor se o índice de início for maior que 0
    }
}
```

```

        if (f->inicio > 0) {
            int j = 0;
            for (int i = f->inicio; i < f->fim; i++, j++) {
                f->itens[j] = f->itens[i];
            }
            f->fim -= f->inicio; // Atualiza o fim
            f->inicio = 0;      // Reinicia o início
        }

        // Após o deslocamento, verifica se há espaço
        if (f->fim == MAX) {
            printf("Fila cheia, não é possível enfileirar\n");
            return false;
        }
    }

    f->itens[f->fim++] = item;
    return true;
}

// Função para desenfileirar um elemento
TipoItem desenfileira(Fila *f) {
    if (filaVazia(f)) {
        printf("Fila vazia, não é possível desenfileirar\n");
        return -1; // Retorno de erro
    }

    TipoItem item = f->itens[f->inicio];
    f->inicio++;
    return item;
}

```

6 - R:

Fila de Pilhas

Uma fila de pilhas é uma estrutura onde cada elemento da fila é uma pilha. A operação de enfileiramento adiciona uma nova pilha ao final da fila, enquanto a operação de desenfileiramento remove a pilha do início da fila. Vou implementar as operações principais: enfileiraPilha e desenfileiraPilha.

C/C++

```

#include <stdio.h>
#include <stdlib.h>

```

```

#define MAX_PILHAS 10
#define MAX_ELEMENTOS 100

typedef struct {
    int itens[MAX_ELEMENTOS];
    int topo;
} Pilha;

typedef struct {
    Pilha pilhas[MAX_PILHAS];
    int inicio, fim;
} FilaDePilhas;

// Função para inicializar a pilha
void inicializaPilha(Pilha *p) {
    p->topo = -1;
}

// Função para empilhar um elemento
void empilha(Pilha *p, int item) {
    p->itens[++p->topo] = item;
}

// Função para inicializar a fila de pilhas
void inicializaFilaDePilhas(FilaDePilhas *f) {
    f->inicio = 0;
    f->fim = 0;
}

// Função para enfileirar uma pilha na fila de pilhas
void enfileiraPilha(FilaDePilhas *f, Pilha p) {
    if ((f->fim + 1) % MAX_PILHAS == f->inicio) {
        printf("Fila de pilhas cheia\n");
        return;
    }
    f->pilhas[f->fim] = p;
    f->fim = (f->fim + 1) % MAX_PILHAS;
}

// Função para desenfileirar uma pilha da fila de pilhas
Pilha desenfileiraPilha(FilaDePilhas *f) {
    if (f->inicio == f->fim) {
        printf("Fila de pilhas vazia\n");
        exit(1);
    }
    Pilha p = f->pilhas[f->inicio];
    f->inicio = (f->inicio + 1) % MAX_PILHAS;
    return p;
}

```

Pilha de Filas

Uma **pilha de filas** é uma estrutura onde cada elemento da pilha é, na verdade, uma fila. A operação de empilhamento adiciona uma nova fila ao topo da pilha, enquanto a operação de desempilhamento remove a fila do topo. Vou implementar as operações principais: **empilhaFila** e **desempilhaFila**.

```
C/C++

#define MAX_FILAS 10
#define MAX_ELEMENTOS_FILA 100

typedef struct {
    int itens[MAX_ELEMENTOS_FILA];
    int inicio, fim;
} Fila;

typedef struct {
    Fila filas[MAX_FILAS];
    int topo;
} PilhaDeFilas;

// Função para inicializar a fila
void inicializaFila(Fila *f) {
    f->inicio = 0;
    f->fim = 0;
}

// Função para enfileirar um elemento na fila
void enfileira(Fila *f, int item) {
    f->itens[f->fim++] = item;
}

// Função para inicializar a pilha de filas
void inicializaPilhaDeFilas(PilhaDeFilas *p) {
    p->topo = -1;
}

// Função para empilhar uma fila na pilha de filas
void empilhaFila(PilhaDeFilas *p, Fila f) {
    if (p->topo == MAX_FILAS - 1) {
        printf("Pilha de filas cheia\n");
        return;
    }
    p->filas[++p->topo] = f;
}

// Função para desempilhar uma fila da pilha de filas
Fila desempilhaFila(PilhaDeFilas *p) {
    if (p->topo == -1) {
        printf("Pilha de filas vazia\n");
        exit(1);
    }
}
```

```

    }
    return p->filas[p->topo--];
}

```

Fila de Filas

Uma **fila de filas** é uma estrutura onde cada elemento da fila é, na verdade, outra fila. A operação de enfileiramento adiciona uma nova fila ao final da fila principal, enquanto a operação de desenfileiramento remove a fila do início da fila principal. Vou implementar as operações principais: **enfileiraFila** e **desenfileiraFila**.

C/C++

```

typedef struct {
    Fila filas[MAX_FILAS];
    int inicio, fim;
} FilaDeFilas;

// Função para inicializar a fila de filas
void inicializaFilaDeFilas(FilaDeFilas *f) {
    f->inicio = 0;
    f->fim = 0;
}

// Função para enfileirar uma fila na fila de filas
void enfileiraFila(FilaDeFilas *f, Fila novaFila) {
    if ((f->fim + 1) % MAX_FILAS == f->inicio) {
        printf("Fila de filas cheia\n");
        return;
    }
    f->filas[f->fim] = novaFila;
    f->fim = (f->fim + 1) % MAX_FILAS;
}

// Função para desenfileirar uma fila da fila de filas
Fila desenfileiraFila(FilaDeFilas *f) {
    if (f->inicio == f->fim) {
        printf("Fila de filas vazia\n");
        exit(1);
    }
    Fila filaRemovida = f->filas[f->inicio];
    f->inicio = (f->inicio + 1) % MAX_FILAS;
    return filaRemovida;
}

```

7 - **R:** pasta: exercício-7

8 - **R:** pasta: exercício-8

9 - **R:** pasta: exercício-9

10 - **R:** pasta: exercício-10