

ATIVIDADE 4 | FILAS E PILHAS

Enos Andrade e Alinne Oliveira

- 1** O que é e como funciona uma estrutura do tipo fila? Em que situações uma fila pode ser utilizada?

Solução

Uma fila é uma estrutura de dados dinâmica que admite remoção de elementos e inserção de novos objetos. Mais especificamente, uma fila é uma estrutura sujeita à seguinte regra de operação: sempre que houver uma remoção, o elemento removido é o que está na estrutura há mais tempo. Em outras palavras, o primeiro objeto inserido na fila é também o primeiro a ser removido. Uma fila pode ser utilizada em filas de impressão e e-commerce (o primeiro comprador a efetuar a compra é o primeiro a ser processado).

- 2** O que é e para que serve uma pilha? Em que situações uma pilha pode ser utilizada?

Solução

Uma pilha é uma estrutura de dados que admite remoção de elementos e inserção de novos objetos. Mais especificamente, uma pilha é uma estrutura sujeita à seguinte regra de operação: sempre que houver uma remoção, o elemento removido é o que está na estrutura há menos tempo. Em outras palavras, o primeiro objeto a ser inserido na pilha é o último a ser removido. Podemos ver exemplos de pilhas em mecanismos de refazer e desfazer dos editores de texto, e navegação entre páginas da web.

- 3** Demonstre como uma sequência de inserções e remoções de uma fila representada por um vetor linear pode provocar estouro ao tentar inserir um elemento numa fila vazia.

Solução

Remoção:

```
int remover(Fila *f){
    if(fila->tamanho == 0){
        printf("Fila vazia\n");
        return;
    }
    int e = f->inicio;
    for(int i = 0; i < f->total - 1; i++){
        f->itens[i] = f->itens[i+1];
    }
    f->total--;
    f->inicio = f->itens[0];
    f->fim = f->itens[f->total];
    return e;
}
```

Inserção:

```
void inserir(Fila *f, int e){
    if(f->total == 0) f->inicio = e;
    if(f->total == 100){
        printf("Fila lotada\n");
        return;
    }
}
```

```
f->fim = e;  
f->itens[f->total] = e;  
f->total++;}
```

- 4** Uma fila recebe os seguintes comandos (em pseudocódigo):

incluir 4, 6, 8, 3, 1

remover 3 elementos

incluir 3, 1, 5, 6

retirar 2 elementos

Que número se encontra na frente da fila?

- (a) 3
- (b) 4
- (c) 5
- (d) 6

Solução

Alternativa (c)

- 5** Uma pilha está inicialmente vazia e depois os seguintes comandos são executados:

push 5

push 7

pop

push 10

push 5

pop

Qual dos seguintes é a pilha correta (suponha que o topo da pilha esteja à esquerda)?

- (a) 5 10 7 5
- (b) 5 10
- (c) 7 5
- (d) 10 5

Solução

Alternativa (d)

- 6** Escreva um módulo `filadeints.c` que implemente uma fila de números inteiros num vetor com redimensionamento. O módulo deve conter as funções `criafile`, `colocanafile`, `tiradafile`, `filavazia`, `liberafile`. (Nessa versão, a função `fila cheia` não faz sentido). Trate os parâmetros da fila como variáveis globais do módulo. Escreva também uma interface `filadeints.h` para o módulo.

Solução

Arquivo `filadeints.h`:

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

#define MAX 100

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

Fila criafila();
void colocanafila(Fila *f, int e);
int tiradafila(Fila *f);
bool filavazia(Fila *f);
void liberafila(Fila *f);
void imprimirfila(Fila *f);
```

Arquivo filadeints.c:

```
#include "filadeints.h"

Fila criafila(){
    Fila fila;
    fila.fim = NULL;
    fila.inicio = NULL;
    fila.total = 0;
    return fila;
}

void colocanafila(Fila *f, int e){
    if(f->total == 0) f->inicio = e;
    if(f->total == 100){
        printf("Fila lotada\n");
        return;
    }
    f->fim = e;
    f->itens[f->total] = e;
    f->total++;
}

int tiradafila(Fila *f){
    if(filavazia(f)){
        printf("Fila vazia\n");
        return;
    }
    int e = f->inicio;
    int i;
    for(i = 0; i < f->total - 1; i++){
        f->itens[i] = f->itens[i+1];
    }
    f->total--;
    f->inicio = f->itens[0];
    f->fim = f->itens[f->total];
    return e;
}

bool filavazia(Fila *f){
```

```
    if(f->total == 0) return true;
    return false;
}
void liberafila(Fila *f){
    f->inicio = f->itens[0];
    f->fim = f->itens[0];
    f->total = 0;
}
void imprimirfila(Fila *f){
    int i;
    for(i = 0; i < f->total; i++){
        printf("%d ", f->itens[i]);
    }
    printf("\n");
}
```

Aplicação - Método main():

```
#include "filadeints.h"

int main() {
    Fila f = criafila();
    colocanafila(&f, 4);
    colocanafila(&f, 6);
    colocanafila(&f, 8);
    colocanafila(&f, 3);
    colocanafila(&f, 1);
    int n1 = tiradafila(&f);
    int n2 = tiradafila(&f);
    int n3 = tiradafila(&f);
    colocanafila(&f, 3);
    colocanafila(&f, 1);
    colocanafila(&f, 5);
    colocanafila(&f, 6);
    int n4 = tiradafila(&f);
    int n5 = tiradafila(&f);
    imprimirfila(&f);
    if(filavazia(&f)) printf("Fila vazia\n");
    else printf("Fila nao vazia\n");
    liberafila(&f);
    printf("Liberando fila...\n");
    if(filavazia(&f)) printf("Fila vazia\n");
    else printf("Fila nao vazia\n");
    liberafila(&f);
    return 0;
}
```

- 7** Escreva um módulo pilhadechars.c que implemente uma pilha de caracteres ASCII. O módulo deve conter as funções criapilha, empilha, desempilha, pilhavazia, pilhacheia. Trate os parâmetros da pilha (o vetor pilha e o índice t) como variáveis globais do módulo. Escreva também uma interface pilhadechars.h para o módulo.

Solução

Arquivo pilhadechars.h:

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

#define MAX 100

typedef struct{
    int qtd;
    char topo;
    char itens[MAX];
} Pilha;

Pilha criapilha();
void empilha(Pilha *p, char e);
char desempilha(Pilha *p);
bool pilhavazia(Pilha *p);
bool pilhacheia(Pilha *p);
void imprimirpilha(Pilha *p);
```

Arquivo pilhadechars.c:

```
#include "pilhadechars.h"

Pilha criapilha(){
    Pilha pilha;
    pilha.qtd = 0;
    pilha.topo = NULL;
    return pilha;
}

void empilha(Pilha *p, char e){
    if(p->qtd == 100){
        printf("Pilha cheia!\n");
        return;
    }
    int i;
    for(i = p->qtd; i > 0; i--){
        p->itens[i] = p->itens[i-1];
    }
    p->topo = e;
    p->itens[0] = e;
    p->qtd += 1;
}

char desempilha(Pilha *p){
    if(p->qtd == 0){
        printf("Pilha vazia!\n");
        return;
    }
    char e = p->topo;
    int i;
    for(i = 0; i < p->qtd - 1; i++){
        p->itens[i] = p->itens[i+1];
    }
    p->topo = p->itens[0];
    p->qtd--;
    return e;
}
```

```
}
bool pilhavazia(Pilha *p){
    if(p->qtd == 0) return true;
    return false;
};
bool pilhacheia(Pilha *p){
    if(p->qtd == 100) return true;
    return false;
};

void imprimirpilha(Pilha *p){
    int i;
    for(i = 0; i < p->qtd; i++){
        printf("%c", p->itens[i]);
    }
    printf("\n");
};
```

```
#include "pilhadechars.h"

int main() {
    Pilha p = criapilha();
    empilha(&p, 'A');
    empilha(&p, 'H');
    empilha(&p, 'L');
    empilha(&p, 'I');
    empilha(&p, 'P');
    imprimirpilha(&p);
    char c1 = desempilha(&p);
    char c2 = desempilha(&p);
    printf("Primeira remocao: %c\n", c1);
    printf("Segunda remocao: %c\n", c2);
    if(pilhavazia(&p)) printf("A pilha esta vazia\n");
    else printf("A pilha nao esta vazia\n");
    if(pilhacheia(&p)) printf("A pilha esta cheia\n");
    else printf("A pilha nao esta cheia\n");
    imprimirpilha(&p);
    return 0;
}
```

- 8** Implemente uma fila dos alunos da disciplina Estrutura de Dados, turno noite, seguindo os mesmos princípios do exercício anterior de listas (Atividade 3), nesse exercício, eu peço que mostre a remoção e depois o seu tamanho, após a remoção do elemento.

Solução

Arquivo filaestudante.h:

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

#define MAX 100
```

```
typedef struct{
    int matricula;
    char *nome;
    float u1;
    float u2;
    float media;
} Estudante;

typedef struct{
    int total;
    Estudante inicio, fim;
    Estudante itens[MAX];
}Fila;

Fila criafila();
void colocanafila(Fila *f, Estudante e);
Estudante tiradafila(Fila *f);
bool filavazia(Fila *f);
void liberafila(Fila *f);
void imprimirfila(Fila *f);
```

Arquivo filaestudante.c:

```
#include <stdio.h>
#include <stdlib.h>
#include "filaestudante.h"

Fila criafila(){
    Fila fila;
    fila.total = 0;
    return fila;
}

void colocanafila(Fila *f, Estudante e){
    if(f->total == 0) f->inicio = e;
    if(f->total == 100){
        printf("Fila lotada\n");
        return;
    }
    f->fim = e;
    f->itens[f->total] = e;
    f->total++;
}

Estudante tiradafila(Fila *f){
    if(filavazia(f)){
        printf("Fila vazia\n");
        return;
    }
    Estudante e = f->inicio;
    int i;
    for(i = 0; i < f->total - 1; i++){
        f->itens[i] = f->itens[i+1];
    }
    f->total--;
    f->inicio = f->itens[0];
    f->fim = f->itens[f->total];
}
```

```
    return e;
}
bool filavazia(Fila *f){
    if(f->total == 0) return true;
    return false;
}
void liberafila(Fila *f){
    f->inicio = f->itens[0];
    f->fim = f->itens[0];
    f->total = 0;
}
void imprimirfila(Fila *f){
    int i;
    for(i = 0; i < f->total; i++){
        printf("\n===== Estudante %d =====\nNome: %s\nMatricula:
        ↪ %d\nMedia: %.1f\n", (i+1), f->itens[i].nome, f->itens[i].matricula,
        ↪ f->itens[i].media);
    }
}
```

```
#include "filaestudante.h"

int main(){
    Fila f = criafila();
    char nomes[15][20] = {"Dennys Angelim", "Diego Barboza", "Ozivan Brito",
    ↪ "Dalison Carvalho", "Lucas Fernandes", "Severino Gomes", "Samaronia
    ↪ Lacerda", "Edberg Martins", "Felipe Maykon", "Alinne Oliveira", "Tamara
    ↪ Ramalho", "Washington Santos", "Joao Victor", "Joao Pedro", "Enos Andrade"};
    float u1[15] = {7.1, 7.5, 7.8, 7.4, 8.8, 8.65, 9.5, 8.1, 7.3, 10, 9, 8.7, 7.5,
    ↪ 8.5, 10};
    float u2[15] = {8, 8.5, 9, 7.5, 7.1, 8.8, 8.5, 7, 7.7, 9.2, 8.3, 8.9, 9.5, 7,
    ↪ 10};
    int i;
    //inclusao de estudates
    for(i = 14; i >= 0; i--){
        Estudante e;
        e.matricula = 2019101 + i;
        e.nome = nomes[i];
        e.u1 = u1[i];
        e.u2 = u2[i];
        e.media = (u1[i] + u2[i]) / 2;
        colocanafila(&f, e);
    }
    imprimirfila(&f);
    printf("\nTamanho antes da remocao: %d\n", f.total);
    Estudante e = tiradafila(&f);
    imprimirfila(&f);
    printf("\n===== Estudante removido =====\nNome: %s\nMatricula:
    ↪ %d\nMedia: %.1f\n", e.nome, e.matricula, e.media);
    printf("\nTamanho depois da remocao: %d\n", f.total);
}
```