Atividade Revisão Estrutura de Dados 2º Bimestre

E-mail *

felipealmeida@esbam.edu.br

Digite seu nome completo: *

Felipe Santos de Almeida

Seção sem título

A estrutura de dados pilha é uma coleção linear onde as inserções e exclusões ocorrem apenas no topo. Na implementação dinâmica de pilhas, a memória é alocada e desalocada conforme a necessidade, o que é vantajoso pois evita o desperdício de memória. A pilha segue a lógica LIFO (Last In, First Out), ou seja, o último elemento inserido é o primeiro a ser removido. Considerando as informações apresentadas, avalie as asserções a seguir e a relação proposta entre elas:

I. Em uma pilha dinâmica, a memória é usada de forma eficiente porque apenas o espaço necessário é alocado.

PORQUE

II. O gerenciamento dinâmico de memória permite a criação de pilhas de tamanho flexível, ajustando-se conforme a necessidade.

A respeito dessas asserções, assinale a opção correta:

* 1 ponto

- (●) As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
- ( ) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
- ( ) A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- ( ) A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- ( ) As asserções I e II são proposições falsas.

Uma fila é uma estrutura de dados linear onde as inserções ocorrem no final e as exclusões ocorrem no início. Diferente da pilha, a fila segue a lógica FIFO (First In, First Out), ou seja, o primeiro elemento inserido é o primeiro a ser removido. Na implementação estática, o tamanho da fila é fixo e predeterminado. Considerando as informações apresentadas, avalie as asserções a seguir e a relação proposta entre elas:

I. A fila estática possui um tamanho fixo que não pode ser alterado durante a execução do programa.

PORQUE

II. A alocação dinâmica de memória é necessária para ajustar o tamanho da fila conforme a necessidade do programa.

A respeito dessas asserções, assinale a opção correta:

* 1 ponto

- [ ] As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
- [x] As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
- [ ] A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- [ ] A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- [ ] As asserções I e II são proposições falsas.

A fila é uma estrutura de dados linear onde as inserções ocorrem no final e as exclusões no início. Em uma fila estática, utilizamos um arranjo de elementos de tamanho predefinido e controlamos a posição do primeiro elemento e o número de elementos na fila. Sobre a fila estática, avalie as afirmações a seguir:

I. A fila estática permite inserções e exclusões em qualquer posição da fila.

II. A posição do próximo elemento a ser inserido é determinada pelo número de elementos na fila.

III. A fila estática utiliza um arranjo de tamanho fixo e predefinido.

IV. Em uma fila estática, o último elemento inserido é sempre o primeiro a ser removido.

É correto o que se afirma em:

* 1 ponto

- ( ) I e II apenas.
- (•) III apenas.
- ( ) III e IV apenas.
- ( ) I, II e III apenas.
- ( ) II, III e IV apenas.

Na implementação dinâmica de pilhas, a memória é alocada e desalocada conforme necessário, o que proporciona flexibilidade no uso de memória. A pilha dinâmica permite a inserção de novos elementos no topo e a remoção também do topo. Sobre a pilha dinâmica, avalie as afirmações a seguir:
I. A pilha dinâmica usa alocação estática de memória.
II. O tamanho da pilha pode crescer e diminuir conforme necessário.
III. Elementos são inseridos e removidos do topo da pilha.
IV. A pilha dinâmica é menos flexível que a pilha estática em termos de gerenciamento de memória.
É correto o que se afirma em:

* 1 ponto

- ( ) I e II apenas.
- (•) II e III apenas.
- ( ) III e IV apenas.
- ( ) I, III e IV apenas.
- ( ) II, III e IV apenas.

A pilha é uma estrutura de dados linear na qual as inserções e exclusões ocorrem no topo. Em uma pilha estática, utilizamos um arranjo de tamanho predefinido e controlamos a posição do elemento no topo da pilha. Qual das alternativas a seguir descreve corretamente uma característica da pilha estática?

* 1 ponto

- ( ) A pilha estática permite inserções em qualquer posição.
- ( ) A pilha estática utiliza alocação dinâmica de memória.
- (•) O elemento no topo da pilha pode ser acessado diretamente, sem remoção.
- ( ) A pilha estática tem um tamanho fixo determinado em tempo de execução.
- ( ) As exclusões na pilha estática ocorrem na base da pilha.

Na implementação de filas estáticas, utilizamos um arranjo de elementos de tamanho predefinido. Controlamos a posição do primeiro elemento e o número de elementos na fila. Qual das alternativas a seguir não é uma característica de uma fila estática? * 1 ponto

- ( ) As inserções ocorrem no final da fila.
- ( ) As exclusões ocorrem no início da fila.
- (•) O tamanho da fila pode ser ajustado dinamicamente.
- ( ) Utiliza um arranjo de tamanho fixo.
- ( ) A fila segue a lógica FIFO (First In, First Out).

Em uma pilha estática, a memória é alocada de forma contínua e seu tamanho é fixo. * 1 ponto
Para inicializar uma pilha estática, devemos:
Complete a frase:
A inicialização de uma pilha estática envolve _____.

- (•) Definir o valor inicial do campo topo como -1.
- ( ) Alocar memória dinamicamente conforme necessário.
- ( ) Definir o valor inicial do campo topo como 0.
- ( ) Estabelecer o tamanho da pilha durante a execução.
- ( ) Definir o valor inicial do campo topo como 1.

Para exibir os elementos de uma fila estática, precisamos iterar pelos elementos válidos e imprimir suas chaves. Suponha que a fila tenha 5 elementos válidos e o primeiro esteja na posição 0 do arranjo. A função de exibição deve:
Complete a frase:
Para exibir os elementos da fila estática, a função deve _____.

* 1 ponto

- ( ) Iterar do final para o início do arranjo.
- (•) Iterar a partir da posição inicial até o número de elementos válidos.
- ( ) Imprimir as chaves dos elementos na ordem inversa.
- ( ) Usar um índice fixo para todas as posições.
- ( ) Imprimir apenas o primeiro e o último elemento.

Em uma fila estática, utilizamos um arranjo de tamanho predefinido e controlamos a posição do primeiro elemento e o número de elementos na fila. A fila segue a lógica FIFO (First In, First Out), o que significa que o primeiro elemento a ser inserido é o primeiro a ser removido. A implementação estática de filas é útil quando sabemos de antemão o número máximo de elementos que a fila pode conter. Com base na descrição acima, responda:
Qual é uma limitação da implementação estática de filas?

* 1 ponto

- (•) O tamanho da fila é fixo e não pode ser alterado durante a execução.
- ( ) As inserções e exclusões ocorrem em qualquer posição do arranjo.
- ( ) A memória é alocada dinamicamente conforme necessário.
- ( ) A fila permite acesso direto a qualquer elemento.
- ( ) Não é possível inicializar a fila após a alocação.

A pilha é uma estrutura de dados linear onde as inserções e exclusões ocorrem no topo. Na implementação dinâmica de pilhas, a memória é alocada e desalocada conforme necessário, evitando o desperdício de memória e permitindo a criação de pilhas de tamanho flexível. Cada elemento na pilha dinâmica indica seu sucessor, formando uma cadeia de elementos. Com base na descrição acima, responda: Qual das seguintes afirmações é verdadeira sobre a pilha dinâmica?

* 1 ponto

- ( ) A pilha dinâmica aloca toda a memória necessária no início da execução.
- ( ) A pilha dinâmica utiliza um arranjo de tamanho fixo.
- (•) A pilha dinâmica permite alocar e desalocar memória conforme necessário.
- ( ) As exclusões ocorrem na base da pilha.
- ( ) A pilha dinâmica não permite inserções após a inicialização.

Na implementação estática de pilhas, utilizamos um arranjo de elementos de tamanho predefinido e controlamos a posição do elemento no topo da pilha. A seguir, temos um exemplo de código em C para a função de inserção (push) em uma pilha estática:
Considerando as informações apresentadas, avalie as asserções a seguir e a relação proposta entre elas:

* 1 ponto

I. A função inserirElementoPilha verifica se a pilha está cheia antes de inserir um novo elemento.
PORQUE
II. A inserção de um elemento em uma pilha estática só ocorre se o índice do topo for menor que o tamanho máximo da pilha menos um.

A respeito dessas asserções, assinale a opção correta:

```c
#define MAX 50

typedef struct {
    int chave;
} REGISTRO;

typedef struct {
    REGISTRO A[MAX];
    int topo;
} PILHA;

void inicializarPilha(PILHA *p) {
    p->topo = -1;
}

bool inserirElementoPilha(PILHA *p, REGISTRO reg) {
    if (p->topo >= MAX - 1) return false;
    p->topo++;
    p->A[p->topo] = reg;
    return true;
}
```

- (●) As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
- ( ) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.

- A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- As asserções I e II são proposições falsas.

Em estruturas de dados, pilhas podem ser implementadas de duas formas: estática e dinâmica. Na implementação estática, utilizamos um arranjo de tamanho fixo, enquanto na implementação dinâmica, alocamos e desalocamos memória conforme a necessidade. A seguir, apresentamos exemplos de código para ambas as implementações.
Sobre as implementações estática e dinâmica de pilhas, avalie as afirmações a seguir: * 1 ponto

I. A pilha estática utiliza um arranjo de tamanho fixo para armazenar os elementos.
II. Na pilha dinâmica, cada elemento aponta para o próximo elemento na pilha, permitindo alocação flexível de memória.
III. A função de inserção na pilha estática retorna falso se a pilha estiver cheia.
IV. Na pilha dinâmica, a função de exclusão libera a memória alocada para o elemento removido.

É correto o que se afirma em:

Implementação Estática:

c

```c
#define MAX 50

typedef struct {
    int chave;
} REGISTRO;

typedef struct {
    REGISTRO A[MAX];
    int topo;
} PILHA;

void inicializarPilha(PILHA *p) {
    p->topo = -1;
}

bool inserirElementoPilha(PILHA *p, REGISTRO reg) {
    if (p->topo >= MAX - 1) return false;
    p->topo++;
    p->A[p->topo] = reg;
    return true;
}

bool excluirElementoPilha(PILHA *p, REGISTRO *reg) {
    if (p->topo == -1) return false;
    *reg = p->A[p->topo];
    p->topo--;
    return true;
}
```

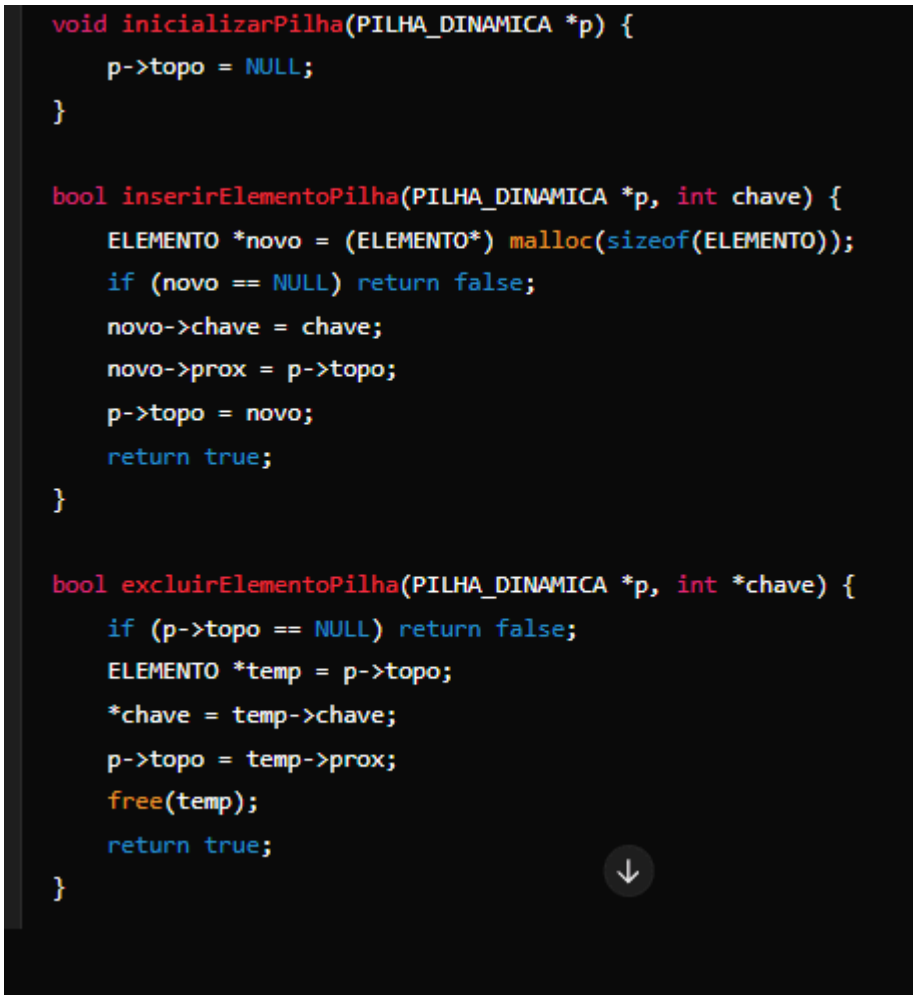
Implementação Dinâmica:

c

```c
#include <stdlib.h>

typedef struct elemento {
    int chave;
    struct elemento *prox;
} ELEMENTO;

typedef struct {
    ELEMENTO *topo;
} PILHA_DINAMICA;
```



```c
void inicializarPilha(PILHA_DINAMICA *p) {
    p->topo = NULL;
}

bool inserirElementoPilha(PILHA_DINAMICA *p, int chave) {
    ELEMENTO *novo = (ELEMENTO*) malloc(sizeof(ELEMENTO));
    if (novo == NULL) return false;
    novo->chave = chave;
    novo->prox = p->topo;
    p->topo = novo;
    return true;
}

bool excluirElementoPilha(PILHA_DINAMICA *p, int *chave) {
    if (p->topo == NULL) return false;
    ELEMENTO *temp = p->topo;
    *chave = temp->chave;
    p->topo = temp->prox;
    free(temp);
    return true;
}
```

- ○ I, II e III apenas.
- ○ I e III apenas.
- ○ II e IV apenas.
- ○ I, II e IV apenas.
- ◉ I, II, III e IV.

Um deque (double-ended queue) é uma estrutura de dados que permite inserções e remoções em ambas as extremidades. A implementação de um deque pode ser realizada utilizando arranjos ou listas ligadas. A seguir, apresentamos um exemplo de implementação de um deque usando uma lista duplamente ligada em C: * 1 ponto

Considere as seguintes operações realizadas sobre um deque:
Inicializar o deque.
Inserir o elemento 10 no início.
Inserir o elemento 20 no fim.
Inserir o elemento 5 no início.
Remover o elemento do fim.
Remover o elemento do início.
Com base no código fornecido e nas operações realizadas, responda:

Qual será o estado final do deque após todas as operações?

c

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

typedef struct elemento {
    int chave;
    struct elemento *prox;
    struct elemento *ant;
} ELEMENTO;

typedef struct {
    ELEMENTO *inicio;
    ELEMENTO *fim;
} DEQUE;

void inicializarDeque(DEQUE *d) {
    d->inicio = NULL;
    d->fim = NULL;
}

bool inserirInicio(DEQUE *d, int chave) {
    ELEMENTO *novo = (ELEMENTO*) malloc(sizeof(ELEMENTO));
    if (novo == NULL) return false;
    novo->chave = chave;
    novo->prox = d->inicio;
    novo->ant = NULL;
    if (d->inicio != NULL) d->inicio->ant = novo;
    d->inicio = novo;
    if (d->fim == NULL) d->fim = novo;
    return true;
}

bool inserirFim(DEQUE *d, int chave) {
    ELEMENTO *novo = (ELEMENTO*) malloc(sizeof(ELEMENTO));
    if (novo == NULL) return false;
    novo->chave = chave;
    novo->ant = d->fim;
    novo->prox = NULL;
    if (d->fim != NULL) d->fim->prox = novo;
    d->fim = novo;
    if (d->inicio == NULL) d->inicio = novo;
    return true;
}

bool excluirInicio(DEQUE *d, int *chave) {
    if (d->inicio == NULL) return false;
    ELEMENTO *temp = d->inicio;
```

```
    *chave = temp->chave;
    d->inicio = d->inicio->prox;
    if (d->inicio != NULL) d->inicio->ant = NULL;
    else d->fim = NULL;
    free(temp);
    return true;
}

bool excluirFim(DEQUE *d, int *chave) {
    if (d->fim == NULL) return false;
    ELEMENTO *temp = d->fim;
    *chave = temp->chave;
    d->fim = d->fim->ant;
    if (d->fim != NULL) d->fim->prox = NULL;
    else d->inicio = NULL;
    free(temp);
    return true;
}
```

- ○ O deque estará vazio.
- ◉ O deque conterá apenas o elemento 10.
- ○ O deque conterá apenas o elemento 5.
- ○ O deque conterá os elementos 5 e 10, nessa ordem.
- ○ O deque conterá os elementos 10 e 20, nessa ordem.

Uma fila é uma estrutura de dados linear onde as inserções ocorrem no final e as exclusões ocorrem no início, seguindo a lógica FIFO (First In, First Out). A seguir, apresentamos um exemplo de código em C que implementa uma fila estática:

* 1 ponto

Com base no código fornecido e nas operações realizadas no main, responda:

Qual será a saída do programa após a execução de todas as operações?

```c
int main() {
    FILA fila;
    int valor;

    inicializarFila(&fila);

    inserirFila(&fila, 10);
    inserirFila(&fila, 20);
    inserirFila(&fila, 30);
    inserirFila(&fila, 40);
    inserirFila(&fila, 50);

    excluirFila(&fila, &valor);
    excluirFila(&fila, &valor);
    inserirFila(&fila, 60);
    inserirFila(&fila, 70);

    while (!filaVazia(&fila)) {
        excluirFila(&fila, &valor);
        printf("%d ", valor);
    }

    return 0;
}
```

- ○ 10 20 30 40 50
- ◉ 30 40 50 60 70
- ○ 50 60 70 10 20
- ○ 60 70 10 20 30
- ○ 10 20 60 70 30

Uma pilha é uma estrutura de dados linear onde as inserções e exclusões ocorrem no topo, seguindo a lógica LIFO (Last In, First Out). A seguir, apresentamos um exemplo de código em C que implementa uma pilha estática: * 1 ponto

Com base no código fornecido e nas operações realizadas no main, responda:

Qual será a saída do programa após a execução de todas as operações?

```c
int main() {
    PILHA pilha;
    int valor;

    inicializarPilha(&pilha);

    inserirPilha(&pilha, 10);
    inserirPilha(&pilha, 20);
    inserirPilha(&pilha, 30);
    inserirPilha(&pilha, 40);
    inserirPilha(&pilha, 50);

    excluirPilha(&pilha, &valor);
    excluirPilha(&pilha, &valor);
    inserirPilha(&pilha, 60);
    inserirPilha(&pilha, 70);

    while (!pilhaVazia(&pilha)) {
        excluirPilha(&pilha, &valor);
        printf("%d ", valor);
    }

    return 0;
}
```

- ○ 10 20 60 70 30
- ○ 70 60 50 40 30
- ◉ 70 60 30 20 10
- ○ 30 40 50 60 70
- ○ 60 70 10 20 30