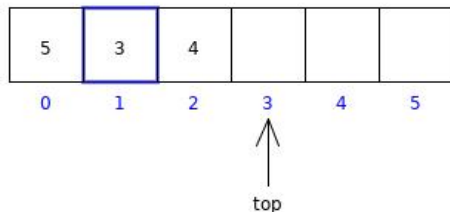


Lista de exercícios

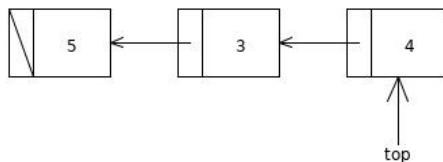
Pilhas

Implementação usando vetores



```
struct TPilha {  
    struct TItem dado[MAX];  
    int Topo;  
};
```

Implementação usando lista encadeada



```
struct Celula {  
    struct TItem Item;  
    struct Celula* Prox;  
}  
  
struct TPilhaDin {  
    struct Celula* top;  
    int qtd;  
};
```

1. Em quais situações uma pilha pode ser utilizada?
2. Implemente as funções push (empilhar) e pop (desempilhar) para as pilhas descritas acima (por vetor e encadeada).
3. Suponha que uma pilha possua 4 valores na seguinte ordem: 1, 2, 3 e 4. Qual seria a sequência correta de operações de inserção (I) e eliminação (E) para se obter os registros na ordem 2 4 3 1?
4. Dada as operações de pilha insere I e remove R, escreva a configuração final da pilha após as seguintes operações: I(10), I(20), R, I(30), I(45), I(21), R, R.
5. Considere uma pilha P vazia e uma fila F não vazia. Utilizando apenas as funções da fila e da pilha, escreva uma função que inverta a ordem dos elementos da fila.
6. Considere uma pilha que armazene caracteres. Faça uma função para determinar se uma string é da forma XY, onde X é uma cadeia formada por caracteres arbitrários e Y é o reverso de X. Por exemplo, se x = ABCD, então y = DCBA. Considere que x e y são duas strings distintas.

7. Desenvolva um método para manter duas pilhas dentro de um único vetor linear (um arranjo) de modo que nenhuma das pilhas incorra em estouro até que toda a memória seja usada, e toda uma pilha nunca seja deslocada para outro local dentro do vetor.

8. Considere que um estacionamento na Rua da Lama, em Vitória, é composto por uma única fileira que guarda até dez carros. Existe apenas uma entrada/saída no estacionamento, e esta extremidade do estacionamento dá acesso justamente à Rua da lama.

Se chegar um cliente para retirar um carro que não seja o mais próximo da saída, todos os carros bloqueando seu caminho sairão do estacionamento. O carro do cliente será manobrado para fora do estacionamento, e os outros carros voltarão a ocupar a mesma sequência inicial.

Escreva um programa que processe um grupo de linhas de entrada. Cada linha de entrada contém um 'E', de entrada, ou um 'S' de saída, e o número da placa do carro.

Presume-se que os carros cheguem e partam na mesma ordem que entraram no estacionamento. O programa deve imprimir uma mensagem sempre que um carro chegar ou sair. Quando um carro chegar, a mensagem deve especificar se existe ou não vaga para o carro no estacionamento. Se não houver vaga, o carro partirá sem entrar no estacionamento. Quando um carro sair do estacionamento, a mensagem deverá incluir o número de vezes em que o carro foi manobrado para fora do estacionamento para permitir que os outros carros saíssem.

Execícios baseados em:

A. Backes. Estrutura de dados descomplicada em linguagem C. Elsevier. 2016.

N. Ziviani, F.C. Botelho, Projeto de Algoritmos com implementações em Java e C++, Editora Thomson, 2006.