**Solução do Teste de Programação: Filas**

- Descrição da Estrutura:

Estrutura:

```
private int size; //Tamanho da pilha
private Node head; //Início da pilha
private int min; //Menor valor armazenado na pilha

//Estrutura de cada nó
private class Node {
    private int number; //Valor armazenado
    private Node next; //Indicação do próximo elemento
}
```

Foi implementada uma estrutura pilha com lista encadeada.

*Por que escolhi a lista encadeada?

Eu prefiro trabalhar com lista encadeada, pois ela apresenta flexibilidade de crescimento e armazenamento da lista.

O nó inicial da lista é sempre o topo da pilha, facilitando o gerenciamento de memória ao aplicar as funções de push e pop, os quais apresentam complexidade constante ao inserir e remover elementos da pilha: O(1).

*Como obter o menor elemento da pilha?

Eu armazenei o menor valor da pilha na estrutura Stack. Toda vez que insiro um elemento na pilha (push), checo se aquele valor é menor do que o meu menor valor armazenado na pilha.

O problema está quando removo (pop) elementos da pilha. Toda vez que chamo a função pop(), chamo internamente a função findMinValue(), com a finalidade de verificar se o elemento removido é o menor valor da pilha.

Essa função findMinValue() apresenta complexidade no pior caso de O(n). Assim, essa função só será chamada no caso de removermos o menor elemento da pilha.

*A complexidade da função min() é O(1).

Apesar dessa solução não apresentar todas as funções com custo computacional constante (O(1)), tendo a função findMinValue() com custo computacional O(n), ela é chamada apenas quando o menor valor da pilha é removido.

Então, o pior caso desse algoritmo é quando removemos o menor elemento da pilha, com complexidade O(n) para identificar o novo menor elemento da pilha.