



Tipo Abstrato de Dados

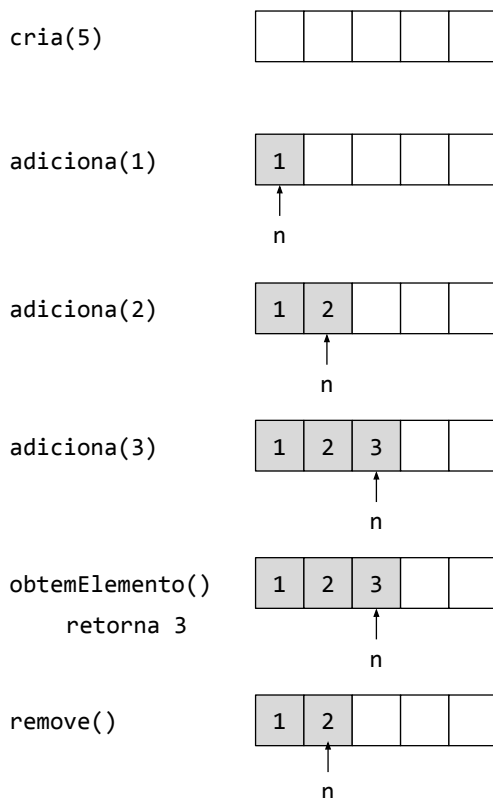
Atenção: Ao terminar, não se esqueça de enviar as soluções no AVA.

TAD Vetor Especial 1 (VE1)

Nesse TAD, sua tarefa é implementar as funções do código VE1.c do arquivo VE1.zip. Diferente de um vetor (array) convencional, o VE1 possui as seguinte propriedade:

- A inserção de um novo elemento é sempre na primeira posição não utilizada;
- Só permite acessar o último elemento;
- O elemento removido é o que está na estrutura há menos tempo, ou seja, o último elemento.

A figura abaixo exemplifica a criação de um VE1 com capacidade para 5 inteiros seguido de algumas operações.



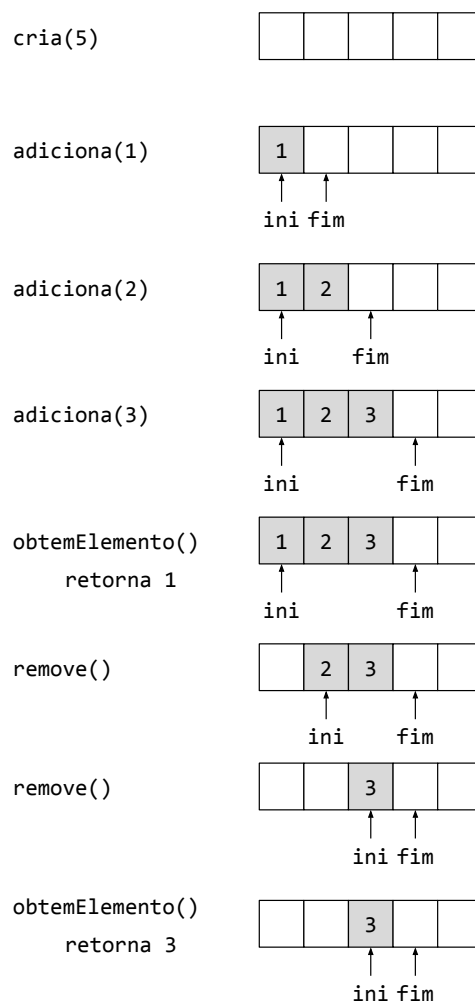
Qual a complexidade de cada função? Preencha no arquivo VE1.c.

TAD Vetor Especial 2 (VE2)

Esse segundo TAD é relativamente parecido com o anterior, mas agora a remoção é sempre do elemento que está há mais tempo na estrutura. O VE2 possui as seguintes propriedades:

- A inserção de um novo elemento é sempre na primeira posição não utilizada;
- Só permite o acesso do “primeiro” elemento;
- O elemento removido é o que está na estrutura há mais tempo, ou seja, o primeiro elemento.

A figura abaixo exemplifica a criação de um VE2 com capacidade para 5 inteiros seguido de algumas operações.

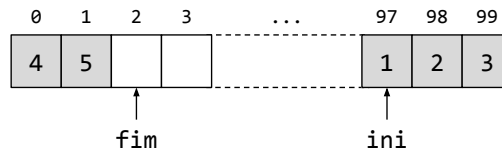


Note que nesse TAD temos dois campos, `ini` e `fim`, que não estavam no TAD anterior. O primeiro (`ini`) indica a posição do “primeiro” elemento (que não necessariamente está na posição 0), quando necessário esse que deve ser removido. Já o campo `fim` indica a posição onde um novo elemento deve ser inserido. Podemos ter também, o campo `n` indicando número de elementos na estrutura.

Como mostrado na figura, observe que os elementos “andam para a direita” dentro do vetor. Isso pode resultar em um problema. Suponha que o tamanho do vetor seja N e que, repetidamente, inserimos e

removemos um único elemento N vezes. Teremos $ini = fim = N$, ou seja, a parte ocupada do vetor chegou à última posição, mesmo tendo várias posições livres.

Para evitar este problema e sermos capazes de utilizar todo o vetor, podemos incrementar as posições do vetor de forma “circular”: se o último elemento ocupa a última posição do vetor, inserimos os novos elementos a partir do início do vetor. Desta forma, em um dado momento, poderíamos ter cinco elementos, 1, 2, 3, 4, 5, distribuídos três no fim do vetor e dois no início (veja a figura abaixo).



Para essa implementação, os índices do vetor são incrementados de maneira que seus valores progridam “circularmente”: cada vez que incrementarmos ini ou fim , simplesmente computamos como $(ini + 1) \% N$ ou $(fim + 1) \% N$.

1. Implemente as funções do código VE2.c do arquivo VE2.zip.
2. Qual a complexidade de cada função? Preencha no arquivo VE2.c.
3. No TAD é utilizado o campo ini para indicar a posição do primeiro elemento, que não necessariamente está na posição 0. O que mudaria se optarmos por não usar esse campo e sempre manter o primeiro elemento na posição 0? A complexidade para inserir e remover um elemento seriam as mesmas? Escreva a sua resposta no fim do arquivo VE2.c.