adicionar elemento

push : 1
aloc : n+1

Se m é a quantidade de elementos imediatamente após a expansão, teremos n push até a próxima expansão. Cada push tem custo 1. Após o vetor alocar novamente, teremos 2m elementos para realocar.
Logo

$$\overbrace{1 . m}^{push} + \overbrace{2m+1}^{realocação} = 3m+1$$

Custo total de uma realocação para extra sendo m a quantidade imediata após a expansão
$\therefore$ $T(n) = 3m$. Seja m a quantidade de elementos novos teremos

$\frac{3m+1}{m} = 3 + \frac{1}{m}$ (*), quanto maior o tamanho do vetor, menor será o valor dessa adição (*), logo o custo amortizado por operação tende a 3 => $O(1)$.

remove elementos

Seja m a quantidade exata de elementos após uma retração

Para que precisemos realocar em um vetor menor, teremos que ter $\frac{1}{4}m$ de elementos, pelo menos

remover : custo 1

realocar : $1 + (m-1)$

logo teremos que fazer

$\frac{3}{4}m$ remoções de custo 1

$\frac{1}{m}$ realocações de custo $1+(m-1)$

Logo

deletou

$$\underbrace{\frac{3}{4}m \cdot 1}_{\text{pops}} + \underbrace{\frac{1}{4}m - 1}_{\text{realocar o que sobrou}} = 4m - 1$$

1+ ..

O custo amortizado por operação

Assim $\frac{4m-1}{m} = 4 - \frac{1}{m} \approx \boxed{4} \Rightarrow O(1)$

Deleção e adição (alternância)

tamanho do vetor: $2m$ com n entradas

$c = 4$

$$c_i = \begin{cases} 0, & \text{se } i \leq 2^{m-1} \\ 2, & \text{se } 2^{m-1} < i \leq n \end{cases}$$

$$CT = 2 \cdot (n - 2^{m-1}) = 2n - 2 \cdot 2^{m-1} =$$

$$= 2n - 2^m$$

removemos e adicionarmos 1 elemento:

$(3 + c) - (1+1) = c+1$ e a quantidade de créditos não diminui