У нас есть 2 случая push:

- Простая вставка элемента в стек
- Увеличение размера стека на 2 и вставка элемента

То есть, если выполняется n операций, то наихудший случай для последней вставки O(n), следовательно, наихудший вариант для n операций $O(n)$.

То есть стоимость i операции или i (для i - 1 выделение памяти + вставка i элемента), или 1 (просто вставка элемента).

Если мы будем делать realloc при размере массива равной степени 2, то:

$$\sum_{i=1}^{n} a_i < n + 2 \cdot n \quad = 3n, \text{где } 2 \cdot n \text{ берется из суммы степеней 2 от 0 до } \log_2 n$$

(количество возможных realloc)

Получается, что амортизированная стоимость не будет превышать 3.

Пусть обычный push = 1, 1 в копилку, а еще 1 в копилку на один из уже содержащихся n/2 элементов.

Когда надо будет вставить n-ый элемент, то у нас будет в копилке $n/2-1$ денег для дополнительных вставок.

$$\phi = 2 \cdot T_{\text{num}} - T_{\text{size}},$$

$$\text{где } T_{\text{num}} = T_{\text{size}} \text{ перед расширением } , T_{\text{size}} = \text{size}$$

Амортизированная стоимость операции:

$$\begin{aligned}
a_{m_i} = t_i + \phi_i - \phi_{i-1} \\
&= t_i + (2 * num_i - size_i) - (2 * num_{i-1} - size_{i-1}) \\
&= t_i + (2 * num_i - 2 * (num_i - 1)) - (2 * (num_i - 1) - ()num_i - 1) \\
&= num_i + 2 - (num_i - 1) \\
&= 3,
\end{aligned}$$

где t_i - фактическая стоимость операции, a_{m_i} - амортизированная стоимость.

У нас возможен случай, когда количество элементов $n/2$ и мы делаем много раз push pop push pop push pop.

Тогда стоимость каждой операции $\theta(n)$. Есллли всего мы сделаем таких операций n, то получим стоимость n операций $\theta(n^2)$.

Стратегия - уменьшать, если $size <= capacity/4$.

Если стек заполнен больше и равен, чем на 1/2, то это индентично тому что выше (расширение). Амортизированная стоимость не превышает

3. Если стек заполнен меньше, чем 1/2, то стек не будет расширен. Тогда амортизированная стоимость:

$$
\begin{aligned}
a_{m_i} = t_i + \phi_i - \phi_{i-1} & \\
&= t_i + (size_i/2 - num_i) - (size_{i-1}/2 - num_{i-1}) \\
&= t_i + (size_i/2 - num_i) - (size_{i-1}/2 - (num_i - 1)) \\
&= 1 + (size_i/2 - num_i) - (size_{i-1}/2 - (num_i - 1)) \quad = 0,
\end{aligned}
$$

Если i операция приводит к сжатию, то фактическая стоимость операции $t_i = num_i + 1$, так как один элемент удаляется и num_i элементов перемещается. Также $size_i/2 = size_{i-1}/4 = num_i + 1$. Тогда амортизированная стоимость операции:

$$
\begin{aligned}
a_i = t_i + \phi_i - \phi_{i-1} & \\
&= (num_i + 1) + (size_i/2 - num_i) - (size_{i-1}/2 - num_{i-1}) \\
&= (num_i + 1) + ((num_i + 1) - num_i) - ((2 * num_i + 2) - (num_i + 1)) \\
&= 1
\end{aligned}
$$

Следовательно, амортизированная стоимость везде ограничена сверху константой, тогда фактическая стоимость $O(n)$.

ЧТО ЕСЛИ ДЕЛАТЬ УМЕНЬШЕНИЕ ПРИ (SIZE <= CAPACITY / 2)?
Пусть есть $n/2$ элементов (если добавить еще один элемент, то размер массива увелится до n). Сделаем $n/2$ операцией push pop pop push push pop pop....

Получится, что вставка увеличит массив, а после второго pop массив уменьшится до $n/2$. Теперь второй push снова приведет к увеличению и тд.. Стоимость расширение и сжатия $\theta(n)$, всего таких операций произойдет $\theta(n)$. Получится, что полная стоимость вот этих n операций будет $O(n^2)$. Тогда амортизированная стоимость одной операции $O(n^2)/n$ это $O(n)$