

Исследование работы контейнера `std::vector` из библиотеки `<vector>`

В своем исследовании я решила рассмотреть несколько функций контейнера `std::vector`, а именно: [`insert`](#), [`erase`](#), [`push_back`](#), [`pop_back`](#) и итератор [`begin`](#).

Функции шли в таком порядке: вставка в конец, вставка в середину, вставка в начало и удаление точно так же. И работало все следующим образом:

```
Size: 12, Capacity: 12
0(000001DD65A4200) 0(000001DD65A4204) 0(000001DD65A4208) 0(000001DD65A420C) 0(000001DD65A4210) 0(000001DD65A4214) 0(000001DD65A4218)
0(000001DD65A421C) 0(000001DD65A4220) 0(000001DD65A4224) 0(000001DD65A4228) 0(000001DD65A422C)
Push back
Size: 13, Capacity: 18
0(000001DD659BE70) 0(000001DD659BE74) 0(000001DD659BE78) 0(000001DD659BE7C) 0(000001DD659BE80) 0(000001DD659BE84) 0(000001DD659BE88)
0(000001DD659BE8C) 0(000001DD659BE90) 0(000001DD659BE94) 0(000001DD659BE98) 0(000001DD659BE9C) 111(000001DD659BEA0)
Push middle
Size: 14, Capacity: 18
0(000001DD659BE70) 0(000001DD659BE74) 0(000001DD659BE78) 999(000001DD659BE7C) 0(000001DD659BE80) 0(000001DD659BE84) 0(000001DD659BE88)
0(000001DD659BE8C) 0(000001DD659BE90) 0(000001DD659BE94) 0(000001DD659BE98) 0(000001DD659BE9C) 0(000001DD659BEA0) 111(000001DD659BEA4)
Push begin
Size: 15, Capacity: 18
888(000001DD659BE70) 0(000001DD659BE74) 0(000001DD659BE78) 0(000001DD659BE7C) 999(000001DD659BE80) 0(000001DD659BE84) 0(000001DD659BE88)
0(000001DD659BE8C) 0(000001DD659BE90) 0(000001DD659BE94) 0(000001DD659BE98) 0(000001DD659BE9C) 0(000001DD659BEA0) 0(000001DD659BEA4) 111(000001DD659BEA8)
Pop back
Size: 14, Capacity: 18
888(000001DD659BE70) 0(000001DD659BE74) 0(000001DD659BE78) 0(000001DD659BE7C) 999(000001DD659BE80) 0(000001DD659BE84) 0(000001DD659BE88)
0(000001DD659BE8C) 0(000001DD659BE90) 0(000001DD659BE94) 0(000001DD659BE98) 0(000001DD659BE9C) 0(000001DD659BEA0) 0(000001DD659BEA4)
Pop middle
Size: 13, Capacity: 18
888(000001DD659BE70) 0(000001DD659BE74) 0(000001DD659BE78) 999(000001DD659BE7C) 0(000001DD659BE80) 0(000001DD659BE84) 0(000001DD659BE88)
0(000001DD659BE8C) 0(000001DD659BE90) 0(000001DD659BE94) 0(000001DD659BE98) 0(000001DD659BE9C) 0(000001DD659BEA0)
Pop begin
Size: 12, Capacity: 18
0(000001DD659BE70) 0(000001DD659BE74) 999(000001DD659BE78) 0(000001DD659BE7C) 0(000001DD659BE80) 0(000001DD659BE84) 0(000001DD659BE88)
0(000001DD659BE8C) 0(000001DD659BE90) 0(000001DD659BE94) 0(000001DD659BE98) 0(000001DD659BE9C)
```

Анализ результатов:

1. Вставка элементов:

- При добавлении одного элемента в конец, размер увеличивается и происходит перераспределение памяти.
- Вставка элементов в середину и начало требует сдвига элементов.

2. Удаление элементов:

- Удаление элементов из конца вектора не требует сдвига, что делает эту операцию более эффективной.
- Удаление из середины и начала требует сдвига элементов.

3. Вместимость:

- Вместимость вектора не уменьшается при удалении элементов.

Сложности:

1. Сложность вставки:

- Вставка в конец (`push_back`) имеет сложность $O(n)$ из-за перераспределения памяти.
- Вставка в середину или начало имеет сложность $O(n)$ из-за необходимости сдвига элементов.

2. Сложность удаления:

- Удаление из конца (**pop_back**) имеет сложность $O(1)$.
- Удаление из середины или начала имеет сложность $O(n)$ из-за необходимости сдвига элементов.

Если функцию (**push_back**) перенести в самый конец в данном примере, тогда мы получим сложность $O(1)$, т.к. перераспределение памяти не потребуется.

```
Size: 12, Capacity: 12
0(000002001A534190) 0(000002001A534194) 0(000002001A534198) 0(000002001A53419C) 0(000002001A5341A0) 0(000002001A5341A4) 0(000002001A5341A8) 0(
000002001A5341AC) 0(000002001A5341B0) 0(000002001A5341B4) 0(000002001A5341B8) 0(000002001A5341BC)

Push middle
Size: 13, Capacity: 18
888(000002001A52BE70) 0(000002001A52BE74) 0(000002001A52BE78) 999(000002001A52BE7C) 0(000002001A52BE80) 0(000002001A52BE84) 0(000002001A52BE88)
0(000002001A52BE8C) 0(000002001A52BE90) 0(000002001A52BE94) 0(000002001A52BE98) 0(000002001A52BE9C) 0(000002001A52BEA0) 0(000002001A52BEA4)

Push begin
Size: 14, Capacity: 18
888(000002001A52BE70) 0(000002001A52BE74) 0(000002001A52BE78) 0(000002001A52BE7C) 999(000002001A52BE80) 0(000002001A52BE84) 0(000002001A52BE88)
) 0(000002001A52BE8C) 0(000002001A52BE90) 0(000002001A52BE94) 0(000002001A52BE98) 0(000002001A52BE9C) 0(000002001A52BEA0) 0(000002001A52BEA4)

Push back
Size: 15, Capacity: 18
888(000002001A52BE70) 0(000002001A52BE74) 0(000002001A52BE78) 0(000002001A52BE7C) 999(000002001A52BE80) 0(000002001A52BE84) 0(000002001A52BE88)
) 0(000002001A52BE8C) 0(000002001A52BE90) 0(000002001A52BE94) 0(000002001A52BE98) 0(000002001A52BE9C) 0(000002001A52BEA0) 0(000002001A52BEA4)
111(000002001A52BEA8)

Pop back
Size: 14, Capacity: 18
888(000002001A52BE70) 0(000002001A52BE74) 0(000002001A52BE78) 0(000002001A52BE7C) 999(000002001A52BE80) 0(000002001A52BE84) 0(000002001A52BE88)
) 0(000002001A52BE8C) 0(000002001A52BE90) 0(000002001A52BE94) 0(000002001A52BE98) 0(000002001A52BE9C) 0(000002001A52BEA0) 0(000002001A52BEA4)

Pop middle
Size: 13, Capacity: 18
888(000002001A52BE70) 0(000002001A52BE74) 0(000002001A52BE78) 999(000002001A52BE7C) 0(000002001A52BE80) 0(000002001A52BE84) 0(000002001A52BE88)
) 0(000002001A52BE8C) 0(000002001A52BE90) 0(000002001A52BE94) 0(000002001A52BE98) 0(000002001A52BE9C) 0(000002001A52BEA0)

Pop begin
Size: 12, Capacity: 18
0(000002001A52BE70) 0(000002001A52BE74) 999(000002001A52BE78) 0(000002001A52BE7C) 0(000002001A52BE80) 0(000002001A52BE84) 0(000002001A52BE88)
0(000002001A52BE8C) 0(000002001A52BE90) 0(000002001A52BE94) 0(000002001A52BE98) 0(000002001A52BE9C)
```

Если мы добавим возможность вставки и удаления нескольких элементов, ситуация будет аналогичная, как и в первом анализе результатов.