Ограничения, которые, я наложил, чтобы достичь результата:

- на вход сортировке подается отсортированный массив;
- pivot каждый раз берется последний элемент;
- размер элемента массива достаточно большой;
- размер массива достаточно большой.

В качестве элемента массива, я взял структуру, размер которой (на моей машине) составляет 32 байта (Листинг 1). Размер сортируемого массива - 100000 элементов.

Листинг 1: Элемент сортируемого массива

```
typedef struct {
  size_t a;
  size_t b;
  size_t c;
  size_t d;
} my_struct_t;
```

Листинг 2: Функция сортировки массива пузырьком

```
#define SWAP(t, a, b) do { t c = a; a = b; b = c; } while (0);

void bubble_sort(check_t arr[], int size) {
  for (int i = 0; i < size - 1; i++) {
    for (int j = 0; j < size - i - 1; j++) {
      if (arr[j].a > arr[j + 1].a) {
        SWAP(my_struct_t, arr[j], arr[j + 1]);
      }
    }
  }
}
```

Листинг 3: Функция быстрой сортировки

```
#define SWAP(t, a, b) do { t c = a; a = b; b = c; } while (0);

void my_qsort(my_struct_t arr[], int start, int stop) {
  if (start >= stop) {
    return;
  }

  int left = start;
  int right = stop;
  check_t cur_el;

  my_struct_t mid = arr[right];

  while (left <= right) {
```

```
15      cur_el = arr[left];
16
17      while (cur_el.a < mid.a) {
18        left += 1;
19        cur_el = arr[left];
20      }
21
22      cur_el = arr[right];
23      while (cur_el.a > mid.a) {
24        right -= 1;
25        cur_el = arr[right];
26      }
27
28      if (left <= right) {
29        SWAP(my_struct_t, arr[right], arr[left]);
30        left += 1;
31        right -= 1;
32      }
33    }
34
35    my_qsort(arr, start,  right);
36    my_qsort(arr, left, stop);
37  }
```

```
gcc main.c -pg -Wall -O0 -lc -o app.exe
./app.exe
gprof app.exe gmon.out -p
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls   s/call   s/call  name
 53.03     12.02    12.02        1    12.02    12.02  my_qsort
 47.19     22.71    10.69        1    10.69    10.69  bubble_sort
  0.00     22.71     0.00        1     0.00     0.00  fill_array
```

Рис. 1: Результаты замеров функций сортировки пузырьком и быстрой сортировки

Данный результат можно объяснить следующими факторами:

- на отсортированном массиве сортировка пузырьком только лишь сравнивает значения, но не переставляет никакие элементы в памяти;
- наоборот, при выбраном pivot быстрая сортировка делает много перемещений элементов массива в памяти;
- элементы массива весят достаточно много (32 байта), их swap в памяти занимает достаточно много времени;

- помимо перестановок в памяти, быстрая сортировка так же делает какие-либо сравнения элементов