

Сравнение сортировок

В этом задании вам надо сравнить время работы разных алгоритмов сортировки на различных вариантах входных данных.

Все алгоритмы надо реализовать в виде функции, принимающей сортируемый массив в качестве параметра (и возможно другие параметры). В скобках указано имя функции, которое надо использовать:

- `selection_sort` — (сортировка выбором максимума/минимума)
- `insertion_sort` — (сортировка вставками)
- `bubble_sort` — (сортировка «пузырьком»)
- `shaker_sort` — «шейкерная» сортировка
- `count_sort` — сортировка подсчётом
- `radix_sort` — поразрядная сортировка
- `merge_sort` — сортировка слиянием
- `quick_sort` — «быстрая» сортировка с выбором опорного элемента в начале сортируемого диапазона
- `quick_sort` — «быстрая» сортировка с выбором случайного опорного элемента

Придумайте разные варианты заполнения массива (например: возрастание, убывание, случайный, «почти» возрастающий, «почти» убывающий, несколько одинаковых значений, фиксированная длина в десятичной записи и проч.). Вероятно, не все варианты массивов разумно тестировать на всех алгоритмах. Пределы применимости алгоритмов определяйте самостоятельно.

```
from random import randint, choice

# SIZE random values
SIZE = 20
x = [randint(-10, 10) for k in range(SIZE)]

# SIZE values from VALUES list
SIZE = 20
VALUES = [2, 3, 5, 8]
x = [choice(VALUES) for k in range(SIZE)]

# generate arrays of different size
for SIZE in 10, 100, 1000, 5000, 10000, 20000, 50000:
    x = [choice(VALUES) for k in range(SIZE)]

# make a few changes in sorted array
x = list(range(1000))
for k in range(10):
    L, R = randint(0, len(x) - 1), randint(0, len(x) - 1)
    x[L], x[R] = x[R], x[L]
```

Для каждого запуска надо измерить время, которое потребовалось для сортировки массива. Делается это так:

```
from random import randint
from time import perf_counter

def selection_sort(x):
    for L in range(len(x), 1, -1):
        ind_max = 0
        for k in range(L):
            if x[k] > x[ind_max]:
                ind_max = k
        x[L - 1], x[ind_max] = x[ind_max], x[L - 1]

for SIZE in 10, 100, 1000, 5000, 10000:
    x = [randint(-10, 10) for k in range(SIZE)]
    start_time = perf_counter()
    selection_sort(x)
    elapsed_time = perf_counter() - start_time
    print(f"SIZE={SIZE}, time={elapsed_time:.3f}")
```