```python
import json
import time
from str_json import LARGE_JSON_STRING
import custom_json


def measure_time(func, *args, n=100000):
    start_time = time.time()
    for _ in range(n):
        func(*args)
    return time.time() - start_time


def performance():

    # Измерим скорости выполнения json.loads
    std_loads_time = measure_time(json.loads, LARGE_JSON_STRING)
    print(f"Standard json.loads time: {std_loads_time:.4f}")

    custom_loads_time = measure_time(custom_json.loads, LARGE_JSON_STRING)
    print(f"Custom custom_json.loads time: {custom_loads_time:.4f}")

    parsed_json = json.loads(LARGE_JSON_STRING)
    custom_parsed_json = custom_json.loads(LARGE_JSON_STRING)
    assert parsed_json == custom_parsed_json

    # Измерим скорости выполнения json.dumps
    std_dumps_time = measure_time(json.dumps, parsed_json)
    print(f"Standard json.dumps time: {std_dumps_time:.4f}")

    custom_dumps_time = measure_time(custom_json.dumps, parsed_json)
    print(f"Custom custom_json.dumps time: {custom_dumps_time:.4f}")

    serialized_json = json.dumps(parsed_json)
    custom_serialized_json = custom_json.dumps(parsed_json)
    assert serialized_json == custom_serialized_json
```

100 тыс раз был обработан json, содержащий 760 строк. Кастомная реализация оказалась быстрее при переводе строки JSON в объект, а при обратном преобразовании медленнее.

```
(.venv) i17m5@i17m5:~/deep-python-course/hse_deep_python_aut_24/10$ python perfomance_test.py
Standard json.loads time: 9.7605
Custom custom_json.loads time: 6.1230
Standard json.dumps time: 0.3707
Custom custom_json.dumps time: 0.6797
```