Эмпирический анализ стандартного алгоритма MERGE SORT

Данные к заданию

```
import matplotlib.pyplot as plt
sizes = [x for x in range(500, 10001, 100)]

random_times = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1,
1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 1, 2,
2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 4, 3, 3, 3, 3, 3, 4, 3, 3, 3, 3, 4,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]

reverse_times = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1,
1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 3, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 4, 4, 3, 4, 4, 4, 5, 4, 4, 3, 3, 3, 3, 3]

nearly_sorted_times = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1,
1,
1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 4, 4, 4, 4, 4, 4, 5, 5, 4, 3, 3, 3, 3, 3, 3]
```

График 1: Построение графика для случайных данных

```
plt.figure(figsize=(10, 6))
plt.plot(sizes, random_times, marker='o', label='Random')
plt.xlabel('Array Size')
plt.ylabel('Time (ms)')
plt.title('Merge Sort - Random Data')
plt.legend()
plt.grid(True)
plt.show()
```

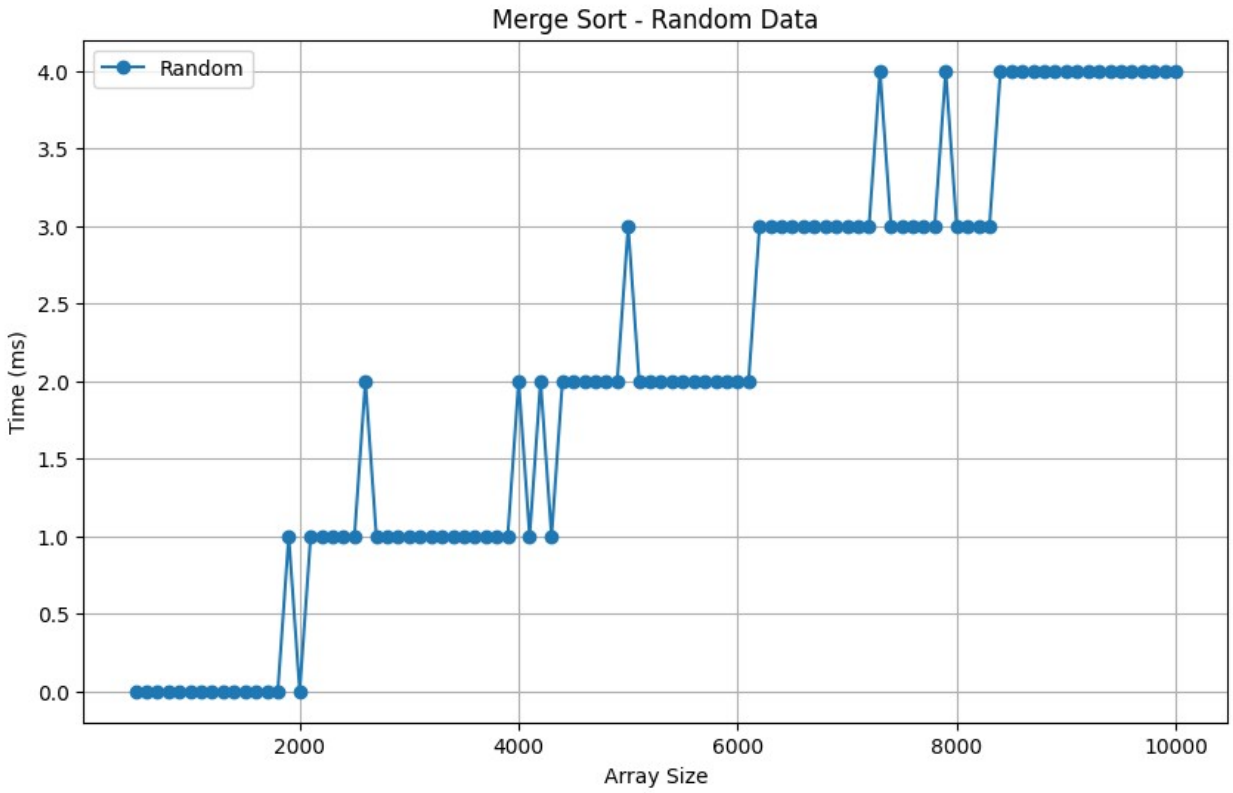


График 2: Построение графика для отсортированных в обратном порядке данных

```
plt.figure(figsize=(10, 6))
plt.plot(sizes, reverse_times, marker='o', label='Reverse Sorted')
plt.xlabel('Array Size')
plt.ylabel('Time (ms)')
plt.title('Merge Sort - Reverse Sorted Data')
plt.legend()
plt.grid(True)
plt.show()
```

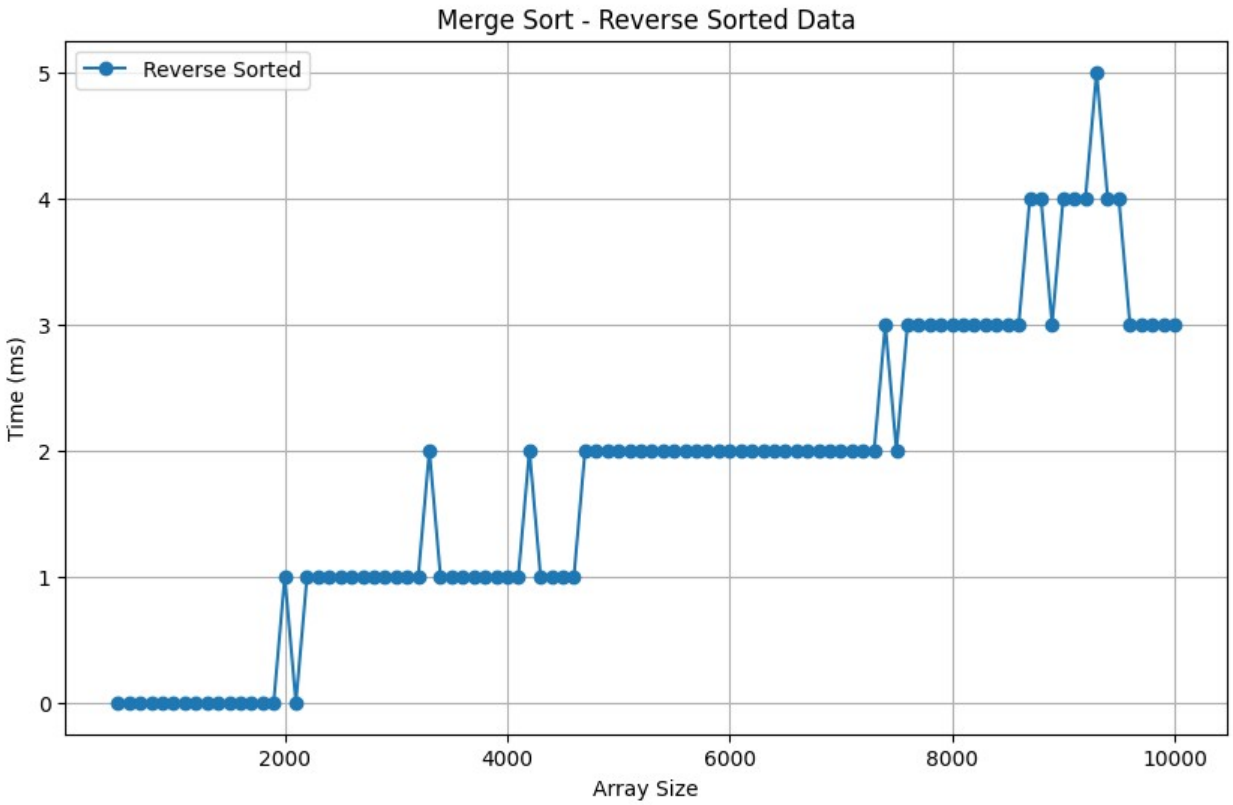
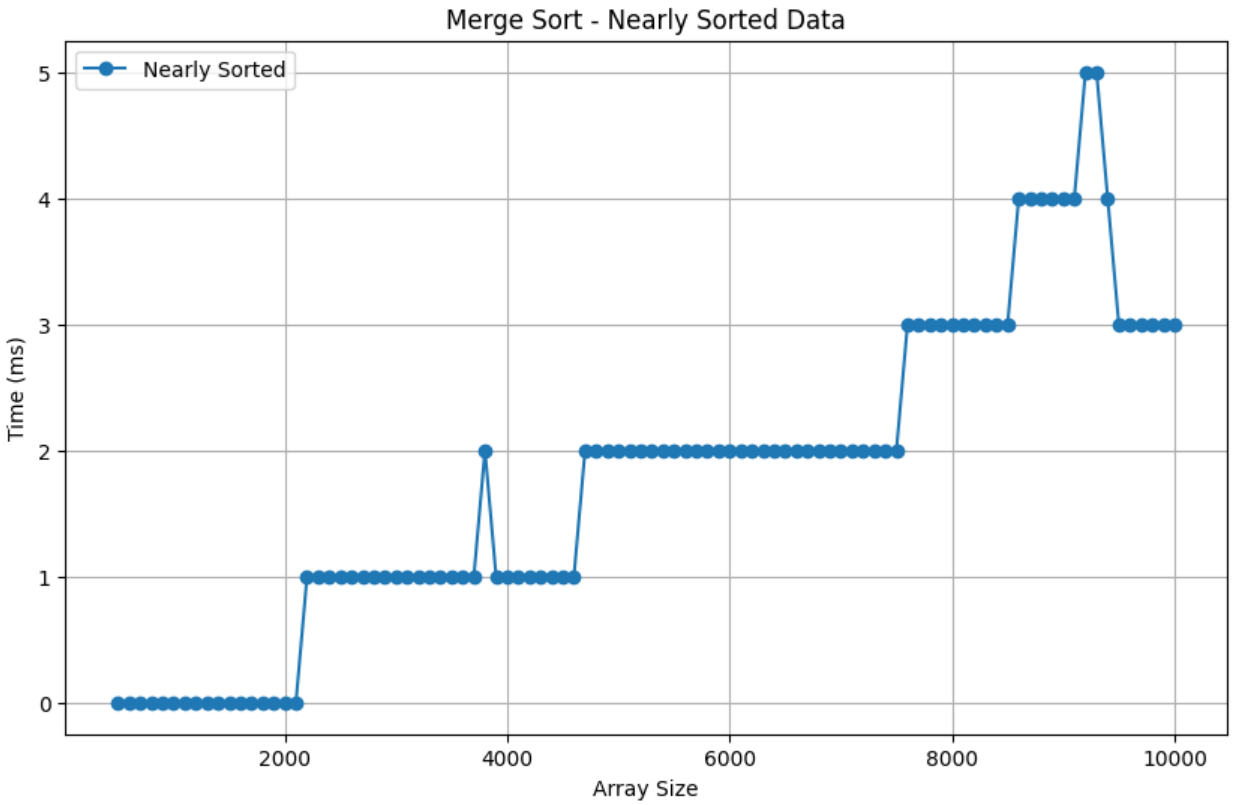


График 3: Построение графика для "почти" отсортированных данных

```
plt.figure(figsize=(10, 6))
plt.plot(sizes, nearly_sorted_times, marker='o', label='Nearly
Sorted')
plt.xlabel('Array Size')
plt.ylabel('Time (ms)')
plt.title('Merge Sort - Nearly Sorted Data')
plt.legend()
plt.grid(True)
plt.show()
```


Merge Sort - Nearly Sorted Data
Nearly Sorted
Time (ms)
Array Size
0
1
2
3
4
5
2000
4000
6000
8000
10000