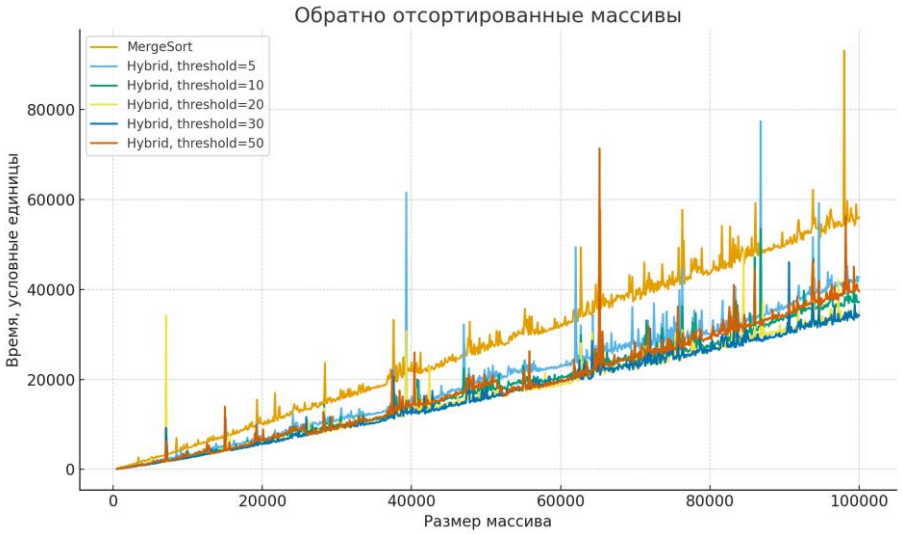
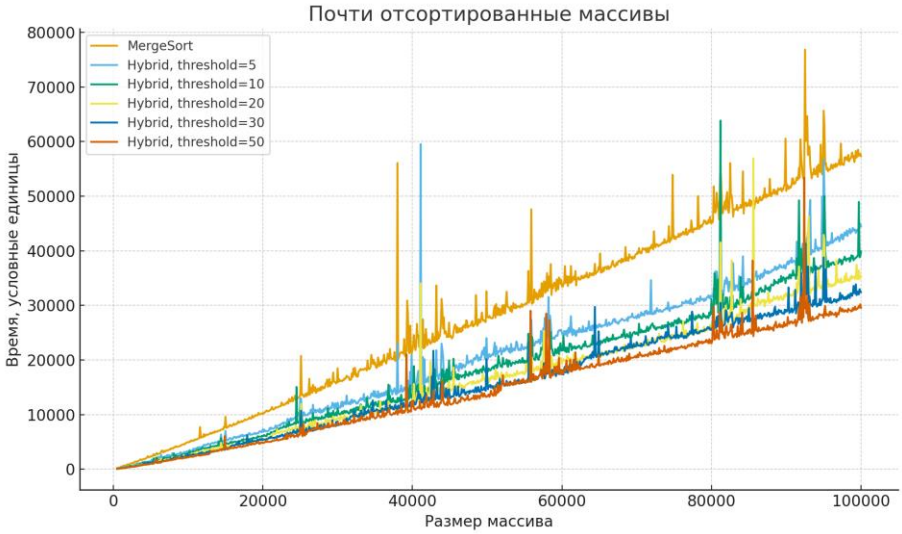
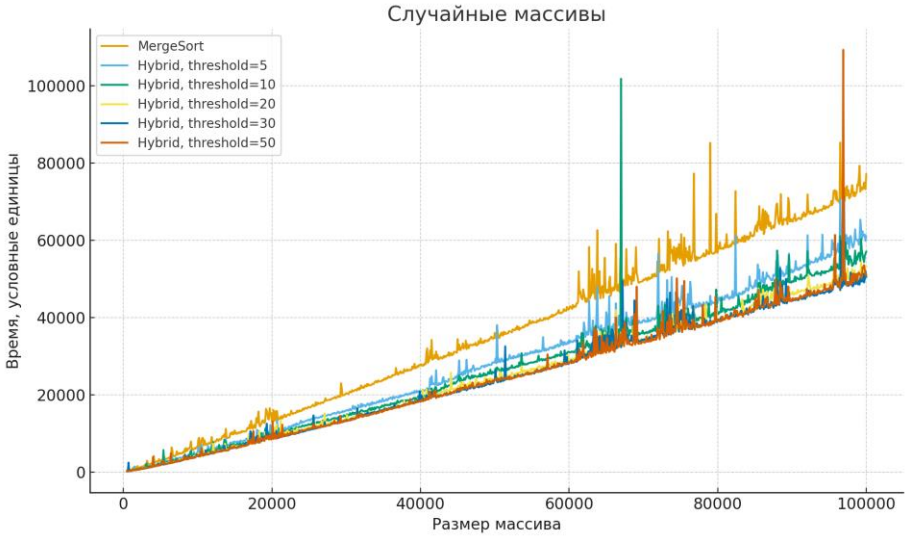
Графики:


Обратно отсортированные массивы
MergeSort
Hybrid, threshold=5
Hybrid, threshold=10
Hybrid, threshold=20
Hybrid, threshold=30
Hybrid, threshold=50
Время, условные единицы
Размер массива


Почти отсортированные массивы
MergeSort
Hybrid, threshold=5
Hybrid, threshold=10
Hybrid, threshold=20
Hybrid, threshold=30
Hybrid, threshold=50
Время, условные единицы
Размер массива


Случайные массивы
MergeSort
Hybrid, threshold=5
Hybrid, threshold=10
Hybrid, threshold=20
Hybrid, threshold=30
Hybrid, threshold=50
Время, условные единицы
Размер массива

Выводы

1. Общие наблюдения

• На всех типах данных чистый MergeSort самый медленный.

• Любая гибридная версия Merge+Insertion даёт ускорение примерно в 1.3–2 раза.

• Рост времени близок к O(n log n), без странных скачков.

2. Случайные массивы

• При N ≥ 50 000 гибрид быстрее MergeSort примерно:

– t=5: ~1.25×, t=10: ~1.35×, t=20: ~1.45×, t=30–50: ~1.47×.

• Оптимальный диапазон порога: 20–30 элементов.

3. Обратно отсортированные массивы

• Для больших N:

– t=5: ~1.4×, t=10: ~1.5×, t=20–30: ~1.65–1.66×, t=50: ~1.5×.

• Слишком большой порог (50) ухудшает результат: InsertionSort на длинных «плохих» кусках слишком дорог

• Разумный порог: 20–30 элементов.

4. Почти отсортированные массивы

• Здесь InsertionSort максимально выгоден: при N ≥ 50 000 гибрид быстрее MergeSort примерно:

– t=5: ~1.4×, t=10: ~1.5×, t=20: ~1.7×, t=30: ~1.8×, t=50: ~1.9×.

• Чем больше threshold, тем лучше: порог 50 даёт лучший результат.

5. Итог

• Гибридный Merge+Insertion практически во всех экспериментах заметно лучше чистого MergeSort.

• Если настраивать порог под конкретный тип:

– случайный и обратно отсортированный: t ≈ 20–30,

– почти отсортированный: t как можно больше (в нашем эксперименте — 50).

• Если нужен один универсальный порог для всех случаев, разумный компромисс — t ≈ 30:

– почти оптимален на случайных,

– оптимален на обратно отсортированных – на почти отсортированных всего немного хуже t=50, но всё равно почти в 1.8 раза быстрее стандартного MergeSort.

Ссылка на реализацию и данные

- ID посылки задачи A1i на Codeforces: 348678167
- Публичный репозиторий с исходными данными и скриптами построения графиков: https://github.com/LeskaProgrammer/algo