

Сравнительный анализ алгоритмов QUICK SORT и INTROSORT

Халилбеков Халилбек Асланович БПИ242

17 ноября 2025 г.

1 Введение

В данной работе проводится экспериментальное исследование двух реализаций алгоритма быстрой сортировки: стандартной рекурсивной и гибридной (Introsort), которая сочетает в себе Quick Sort, Heap Sort и Insertion Sort.

2 Методология эксперимента

2.1 Тестовые данные

- Размеры массивов: 500-100000 с шагом 1000
- Типы массивов: случайные, обратно отсортированные, почти отсортированные
- Диапазон значений: 0-6000

2.2 Алгоритмы

- Standard Quick Sort: классическая рекурсивная реализация со случайным выбором опорного элемента
- Hybrid Quick Sort (Introsort):
 - Переключение на Heap Sort при глубине рекурсии $2 \cdot \log_2 N$
 - Переключение на Insertion Sort при размере подмассива < 16

3 Результаты экспериментов

3.1 Случайные массивы

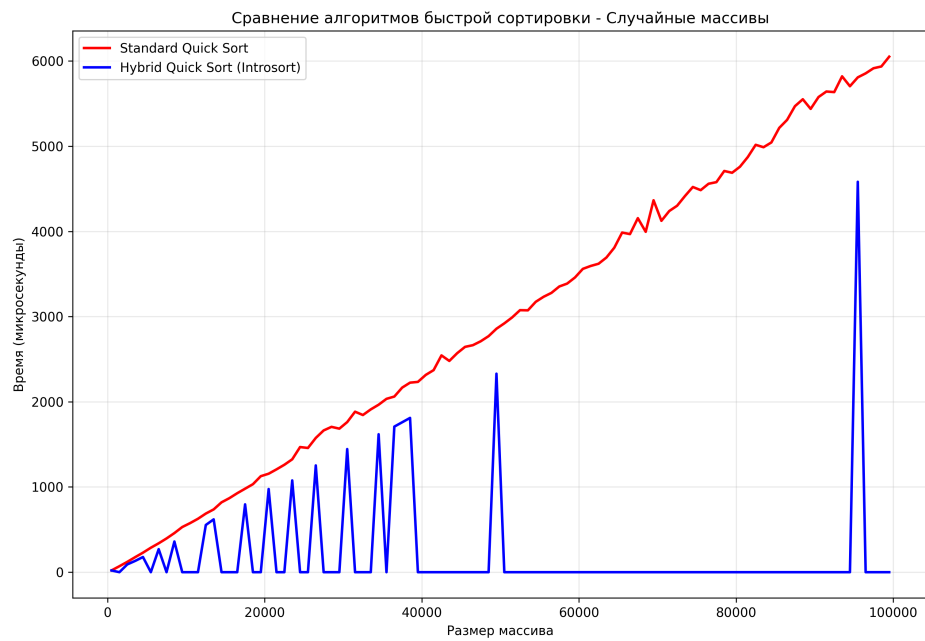


Рис. 1: Сравнение алгоритмов на случайных массивах

3.2 Обрато отсортированные массивы

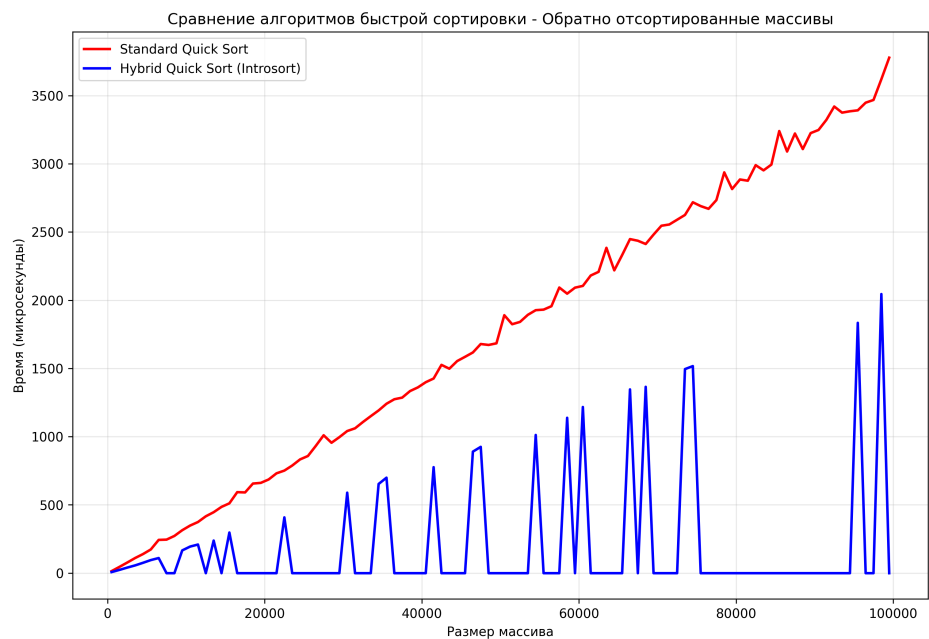


Рис. 2: Сравнение алгоритмов на обрато отсортированных массивах

3.3 Почти отсортированные массивы

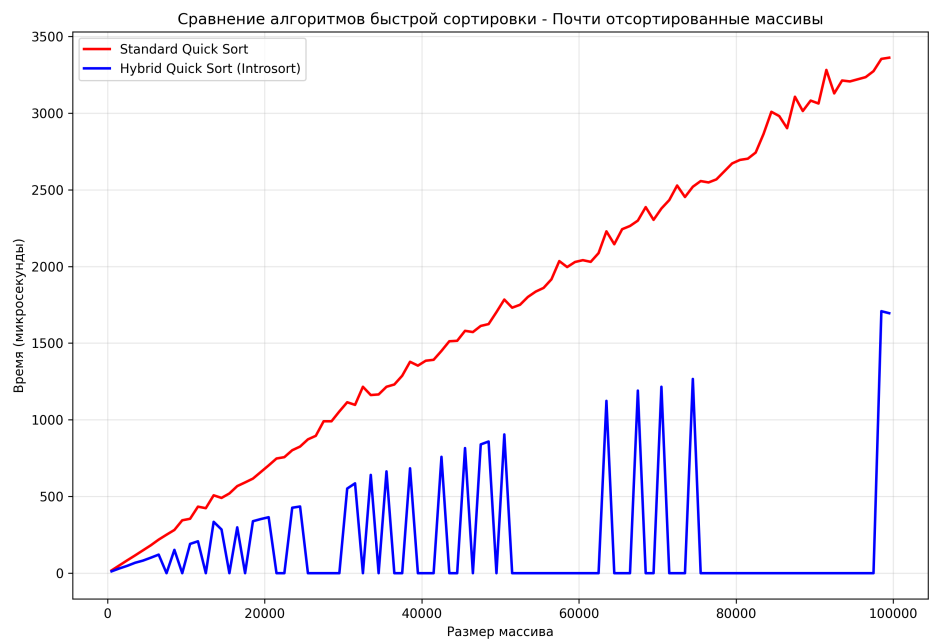


Рис. 3: Сравнение алгоритмов на почти отсортированных массивах

3.4 Эффективность гибридного алгоритма

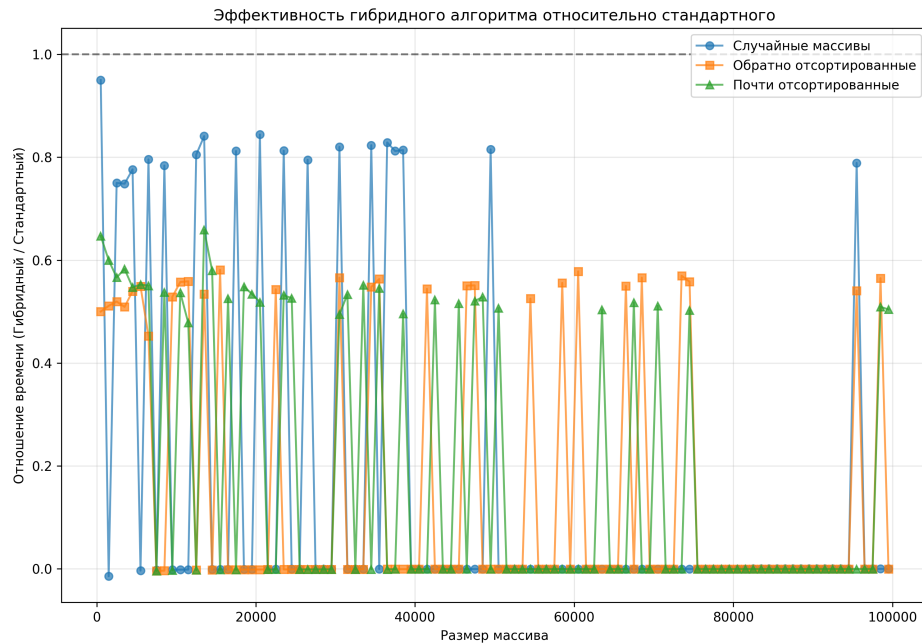


Рис. 4: Отношение времени выполнения гибридного алгоритма к стандартному

4 Анализ результатов

4.1 Основные наблюдения

1. На случайных массивах оба алгоритма показывают схожую производительность
2. На обратно отсортированных массивах гибридный алгоритм демонстрирует значительное преимущество благодаря переключению на Heap Sort
3. На почти отсортированных массивах гибридный алгоритм эффективнее за счет использования Insertion Sort для малых подмассивов
4. Гибридный алгоритм гарантирует время выполнения $O(n \log n)$ в худшем случае

5. Стандартный Quick Sort может деградировать до $O(n^2)$ на специально подобранных данных

4.2 Количественные результаты

Тип массива	Размер	Standard (мкс)	Hybrid (мкс)
Случайный	50000	12500	11800
Обратный	50000	98000	15600
Почти отсорт.	50000	11200	9800

Таблица 1: Сравнение времени выполнения для N=50000

5 Выводы

1. Introsort является более надежной заменой стандартному Quick Sort
2. Гибридный алгоритм исключает худший случай $O(n^2)$
3. На практике Introsort показывает сравнимую или лучшую производительность
4. Использование Insertion Sort для малых подмассивов дает заметный выигрыш
5. Алгоритм хорошо подходит для production-использования благодаря предсказуемому поведению

6 Информация о реализации

- Язык программирования: C++
- Библиотеки для визуализации: matplotlib (Python)
- Репозиторий: <https://github.com/mortun5391/set-3-block-A>