



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И
УПРАВЛЕНИЯ

ЛАБОРАТОРНАЯ РАБОТА № 7.1

**«Сравнительный анализ эффективности алгоритмов сортировки,
поиска и поиска ближайших точек»**

ДИСЦИПЛИНА: «Основы программирования»

Выполнил: студент гр. ИУ5-14Б

Корнеев Г. И.
(Подпись) (Ф.И.О.)

Проверил:

Колосов М. И.
(Подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

2025 г.

Цель работы

Выполнить сравнение временной эффективности различных алгоритмов и представить результаты в графической форме для наглядного анализа их производительности.

Результат работы программы в терминале представлен на рисунке 1

```
korneev@Xiaomi:~/lab7-part1$ rm -rf build && rm -rf csv && cmake -B build && cmake --build build && ./build/sorting_s  
ct && find . -name "*.csv" -type f -exec sed -i 's;/,/g' {} +  
Лабораторная работа 7-1: Сортировки, поиски, ближайшая пара  
  
- СОРТИРОВКИ -  
Выбором | отсортировано=да | время=1.026 мс  
Пузырьком | отсортировано=да | время=6.355 мс  
Слиянием | отсортировано=да | время=0.101 мс  
Совпадение (слияние vs выбор): да  
Совпадение (пузырёк vs выбор): да  
  
- ПОИСК -  
Последовательный: есть=666, нет=-1  
Бинарный (на отсорт.) : есть=435  
  
- БЛИЖАЙШАЯ ПАРА -  
Перебор: i=1047 j=1262 dist=4.47214 t=1.738 мс  
D&C : i=1047 j=1262 dist=4.47214 t=0.771 мс  
CSV в папке ./csv: sorting.csv, sorting_cases.csv, sorting_sizes.csv, search.csv, search_sizes.csv, closest_pair.csv
```

Рисунок 1

Таблица результатов представлена на рисунке 2

A	B	C
Algorithm	Case	Time_ms
SelectionSort	Лучший	0.238000000
BubbleSort	Лучший	0.001000000
MergeSort	Лучший	0.004000000
SelectionSort	Средний	0.269000000
BubbleSort	Средний	1.633000000
MergeSort	Средний	0.040000000
SelectionSort	Худший	0.244000000
BubbleSort	Худший	2.924000000
MergeSort	Худший	0.011000000

Рисунок 2

Поиск ближайших точек

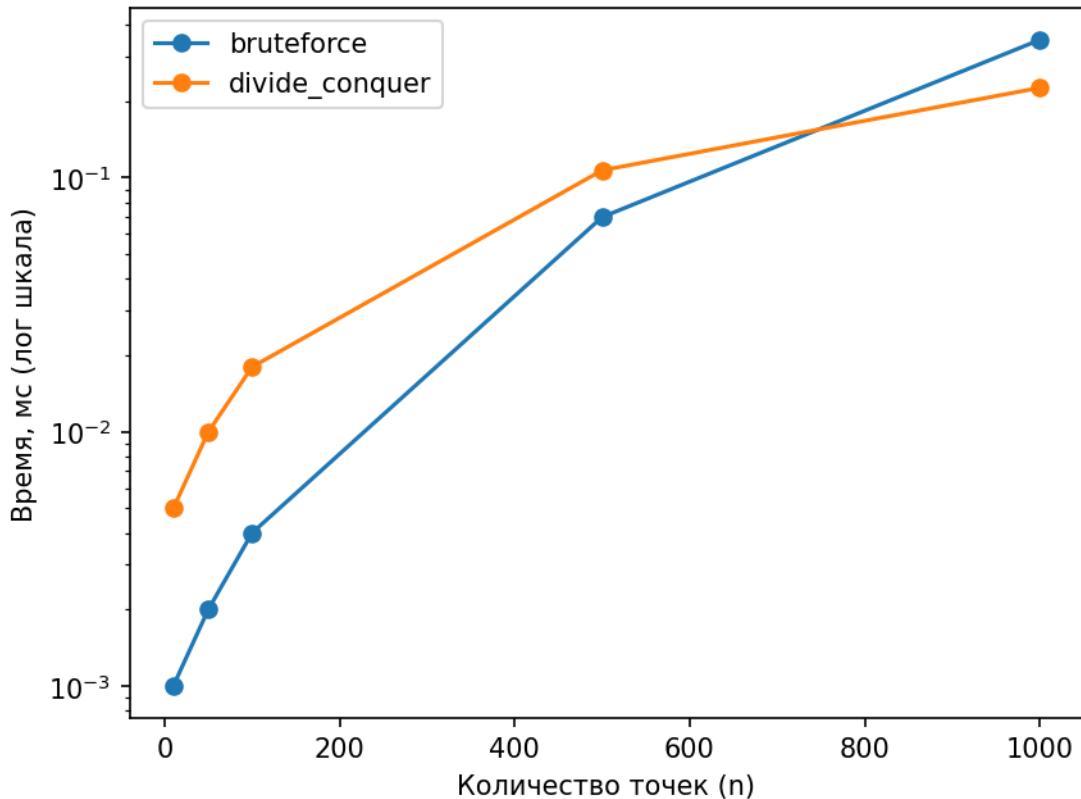


Рисунок 3

Таблица Сортировка (разные размеры массива)

	A	B	C
1	n	method	time_ms
2	100	selection	0.008000000
3	100	bubble	0.019000000
4	100	merge	0.004000000
5	300	selection	0.034000000
6	300	bubble	0.177000000
7	300	merge	0.015000000
8	500	selection	0.063000000
9	500	bubble	0.436000000
10	500	merge	0.019000000
11	1000	selection	0.270000000
12	1000	bubble	1.750000000
13	1000	merge	0.043000000
14	2000	selection	0.912000000
15	2000	bubble	6.656000000
16	2000	merge	0.099000000
17	5000	selection	4.595000000
18	5000	bubble	40.778000000
19	5000	merge	0.229000000

Рисунок 4

Таблица Времени поиска ближайших точек при разном количестве точек

	A	B	C
1	n	method	time_ms
2	10	bruteforce	0.001000000
3	10	divide_conquer	0.005000000
4	50	bruteforce	0.002000000
5	50	divide_conquer	0.010000000
6	100	bruteforce	0.004000000
7	100	divide_conquer	0.018000000
8	500	bruteforce	0.070000000
9	500	divide_conquer	0.107000000
10	1000	bruteforce	0.349000000
11	1000	divide_conquer	0.226000000

Рисунок 5

Выводы

Проведённый эксперимент подтвердил, что сортировка слиянием и метод декомпозиции обеспечивают наилучшую эффективность при увеличении объёма данных, тогда как простые алгоритмы с квадратичной сложностью быстро теряют производительность. Бинарный поиск также значительно превосходит последовательный при работе с большими массивами. Полученные результаты визуально подтверждают теоретические оценки сложности и демонстрируют практическое преимущество более оптимальных алгоритмов.

Заключение

Эксперимент подтвердил, что алгоритмы с меньшей асимптотической сложностью, такие как сортировка слиянием и метод декомпозиции, обеспечивают наилучшую производительность при увеличении объёма данных. Полученные результаты совпадают с теоретическими оценками и подчёркивают важность выбора эффективного алгоритма для конкретных задач.