



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 № 7.2

«Поиск самой длинной возрастающей подпоследовательности
(LIS)»

ДИСЦИПЛИНА: «Основы программирования»

Выполнил: студент гр. ИУ5-14Б

(Подпись) Корнеев Г. И.
(Ф.И.О.)

Проверил:

(Подпись) Колосов М. И.
(Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

2025 г.

Цель работы

Исследование алгоритмов поиска самой длинной возрастающей подпоследовательности (LIS), сравнение их вычислительной сложности и практической эффективности на наборах данных различного размера.

Тема работы

Сравнительный анализ алгоритмов поиска LIS: наивного алгоритма и алгоритма динамического программирования $O(n^2)$. Экспериментальное определение вычислительной сложности.

Для решения задачи поиска самой длинной возрастающей подпоследовательности был выбран метод динамического программирования. Он подходит лучше всего, потому что задача имеет оптимальную подструктуру: решение для позиции i напрямую зависит от решений для всех предыдущих элементов. Подзадачи перекрываются, поэтому хранение промежуточных результатов существенно ускоряет вычисления. Метод декомпозиции и уменьшения размера задачи здесь неприменимы, так как подпоследовательность может пересекать границы разбиения и не раскладывается на независимые части. Динамическое программирование даёт простую и прозрачную реализацию с предсказуемой сложностью $O(n^2)$.

Структура проекта в vs code

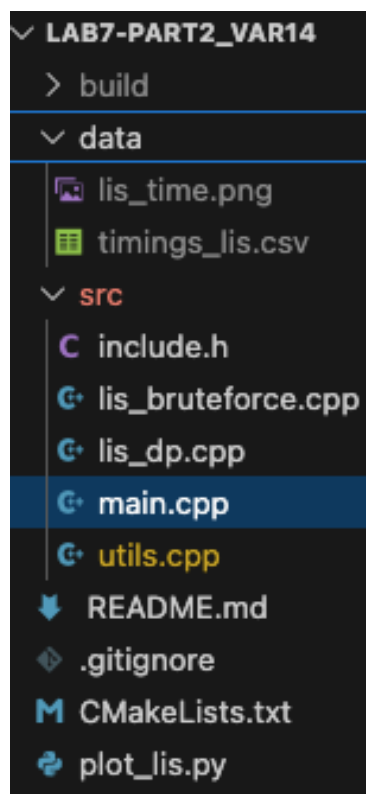


Рисунок 1

Результат работы программы в VS Code на рисунке 2

```
[100%] Linking CXX executable bin/app
[100%] Built target app
Размер массива n = 50    Время = 0 мс    Длина LIS = 11
Размер массива n = 100   Время = 0 мс    Длина LIS = 17
Размер массива n = 200   Время = 0 мс    Длина LIS = 22
Размер массива n = 400   Время = 1 мс    Длина LIS = 33
Размер массива n = 800   Время = 5 мс    Длина LIS = 52
Размер массива n = 1200  Время = 17 мс   Длина LIS = 65
Размер массива n = 1600  Время = 32 мс   Длина LIS = 77
Размер массива n = 2000  Время = 37 мс   Длина LIS = 84
Размер массива n = 5000  Время = 199 мс  Длина LIS = 134
Размер массива n = 10000 Время = 1320 мс Длина LIS = 197
Размер массива n = 20000 Время = 4160 мс Длина LIS = 281
Размер массива n = 40000 Время = 14448 мс Длина LIS = 383

Демонстрационный пример:
Массив: 10 9 2 5 3 7 101 18
Длина LIS: 4
Подпоследовательность: 2 5 7 101

CSV-файл сохранён: data/timings_lis.csv
```

Рисунок 2

Для метода ДП были проведены замеры на массивах размеров 50–40000. Для каждого размера выполнялось по 5 прогонов на случайных данных. Результаты сохранялись в CSV-файл, который затем использовался для построения графика

Вывод в csv таблице представлен на рисунке 3

	A	B	C	D
1	method	n	time_ms	length
2	bruteforce	10	0	5
3	dp	10	0	5
4	bruteforce	12	0	6
5	dp	12	0	6
6	bruteforce	15	1	6
7	dp	15	0	6
8	bruteforce	18	11	5
9	dp	18	0	5
10	dp	50	0	11
11	dp	100	0	17
12	dp	200	0	22
13	dp	400	1	33
14	dp	800	5	52
15	dp	1200	17	65
16	dp	1600	32	77
17	dp	2000	37	84
18	dp	5000	199	134
19	dp	10000	1320	197
20	dp	20000	4160	281
21	dp	40000	14448	383

Рисунок 3

По данным CSV был построен график зависимости времени выполнения от размера массива. График показывает квадратичный рост, что визуально подтверждает теоретическую сложность алгоритма. График сравнения методов представлен на рисунке 4

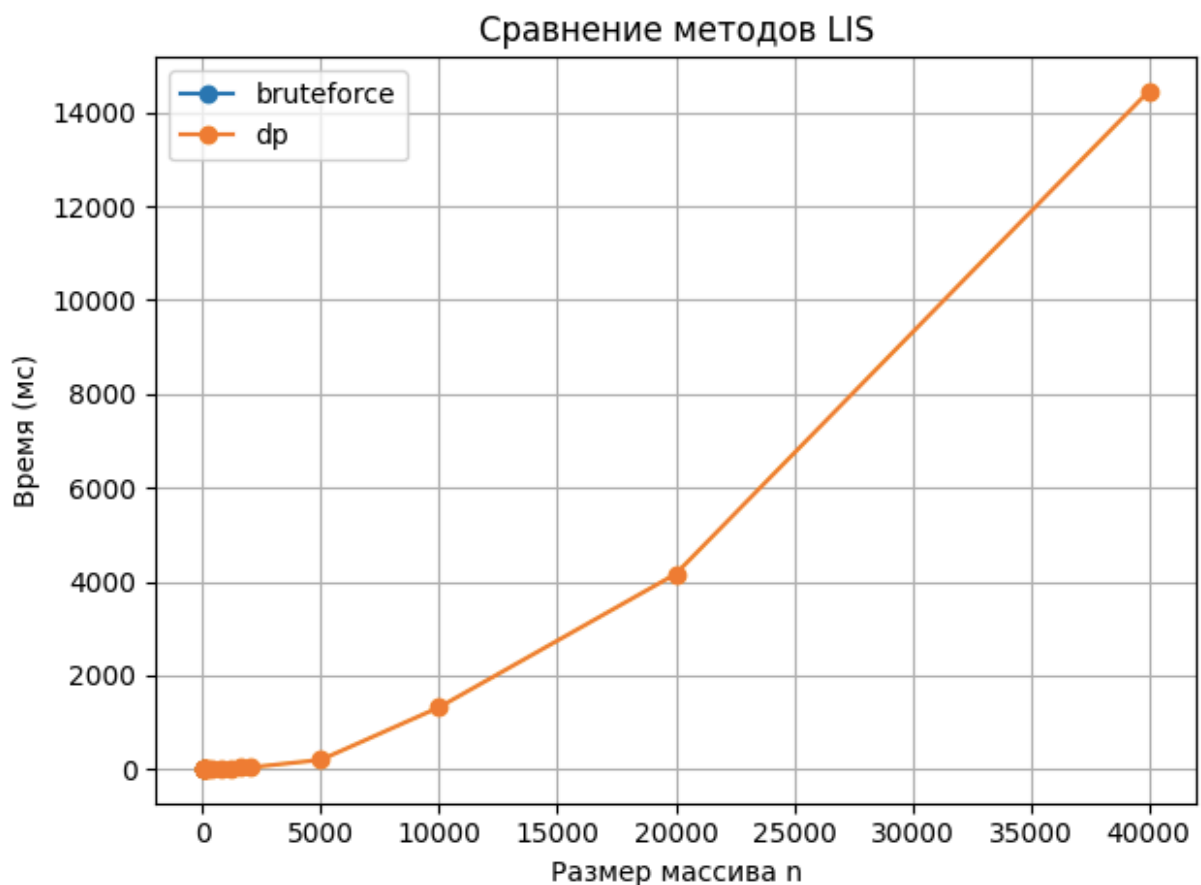


Рисунок 4

ИИ в этой работе использовался для оперативной отладки ошибок и проверки корректности получаемых данных.

```

исходный массив: 10 9 2 5 3 7 10 1 18
Длина LIS = 4
Последовательность: 2 5 7 10 1

CSV сохранён в data/timings_lis.csv

Отлично, всё собралось и отработало как нужно — лог показывает ожидаемую картину для  $O(n^2)$ :
время почти квадратично растёт с  $n$  ( $\approx 0$  ms до 400, затем 2–3 ms на 800,  $\sim 7$  ms на 1200,  $\sim 13$ –30
  
```

Рисунок 5

Вывод

В ходе работы был исследован алгоритм поиска самой длинной возрастающей подпоследовательности. Реализованный метод динамического программирования показал корректность и предсказуемую квадратичную сложность $O(n^2)$, что полностью подтвердилось экспериментальными измерениями. Альтернативный наивный алгоритм позволил сверить результаты и продемонстрировал резкое ухудшение производительности при увеличении размера входных данных. Построенный график наглядно показал характер роста времени

выполнения и соответствие теоретической оценке. Проведённое исследование подтвердило, что метод динамического программирования является оптимальным выбором для решения задачи в рамках лабораторной работы.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Быков, А. Ю. Решение задач на языках программирования Си и Си++ : методические указания / А. Ю. Быков. — Москва : МГТУ им. Н.Э. Баумана, 2017. — 248 с. — ISBN 978-5-7038-4577-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/103505>
2. Каширин, И. Ю. От Си к Си++ : учебное пособие / И. Ю. Каширин, В. С. Новичков. — 2-е изд., стер. — Москва : Горячая линия-Телеком, 2012. — 334 с. — ISBN 978-5-9912-0259-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/5161>
3. Быков А. Ю. Решение задач на языках программирования Си и Си++ : метод. указания к выполнению лаб. работ / Быков А. Ю. ; МГТУ им. Н. Э. Баумана. - М. : Изд-во МГТУ им. Н. Э. Баумана, 2017. - 244 с. : ил. - ISBN 978-5-7038-4577-6.
4. Иванова Г. С., Ничушкина Т. Н. Объектно-ориентированное программирование : учебник для вузов / Иванова Г. С., Ничушкина Т. Н. ; общ. ред. Иванова Г. С. - М. : Изд-во МГТУ Н.Э.Баумана, 2002. <http://progbook.ru/technologiya-programmirovaniya/582-ivanova-tehnologiya-programmirovaniya.html>
5. Иванова Г. С., Ничушкина Т. Н. Объектно-ориентированное программирование : учебник для вузов / Иванова Г. С., Ничушкина Т. Н. ; общ. ред. Иванова Г. С. - М. : Изд-во МГТУ им. Н. Э. Баумана, 2014. - 455 с. : ил. - Библиогр.: с. 450. - ISBN 978-5-7038-3921-8.
6. Подбельский В.В. Язык Си++: Учебное пособие. – М.: Финансы и статистика, 2003. <http://progbook.ru/c/737-podbelskii-programmiovanie-na-yazyke-si.html>.
7. Акулов О.А., Медведев Н.В. Информатика. Базовый курс. – М.: Омега-Л, 2006. <http://razym.ru/naukaobraz/obrazov/151874-akulov-oa-medvedev-nv-informatika-bazovyy-kurs.html>
8. Вычислительные методы и программирование. МГУ им. М.В. Ломоносова. ISSN 1726-3522. Журнал входит в 1-й уровень Белого списка

Дополнительные материалы

1. Иванова Г.С. Технология программирования: Учебник для вузов. – М.: Изд. МГТУ им. Ахо А.В., Хопкрофт Д.Э., Ульман Д.Д. Структуры данных и алгоритмы. – М., Вильямс, 2003. <http://razym.ru/naukaobraz/obrazov/181547-aho-a-ulman-d-hopkroft-d-struktury-dannyh-i-algoritmy.html>
2. Дейтел Х.М., Дейтел П.Дж. Как программировать на C++. – М.: Бином, 2001.
3. Т. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М. МЦНМО, 2005. <http://rutracker.org/forum/viewtopic.php?t=533181>
4. Джосьютис Н. C++ Стандартная библиотека для профессионалов. – СПб.: Питер, 2004. http://progbook.ru/c/178-dzhosyutis_c_standartnaya_biblioteka.html
5. Подбельский В.В. Стандартный Си++: Учебное пособие. – М.: Финансы и статистика, 2008.
6. Объектно-ориентированное программирование в C++: пер. с англ. / Лафоре Р. - 4-е изд. - СПб.: Питер, 2004. - 923 с. - (Классика computer science). - ISBN 5-94723-302-9.
7. Т. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М. МЦНМО, 2005.
8. Г. Шилдт. C++. Базовый курс, 3-е издание: Пер с англ. – М.: Издательский дом «Вильямс», 2011. – 624 с.
9. Павловская Т. А. C/C++. Программирование на языке высокого уровня: учебник для вузов / Павловская Т. А. - СПб.: Питер, 2003. - 460 с. - (Учебник для вузов). - ISBN 5-94723-568-4.
10. Бесплатные образовательные программы партнера (VK): <https://education.vk.company/students>