Учреждение образования

«Белорусский государственный технологический университет»

**Отчет по лабораторной работе №4**

**ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ**

Выполнил:

Студент ФИТ 2-7-2

Лешук Дмитрий

**ЦЕЛЬ РАБОТЫ:** освоить общие принципы решения задач методом динамического программирования, сравнить полученные решения задач с рекурсивным методом.

**Задание №1)** Генерация строк

Для генерации строки, содержащей случайные символы, воспользуемся следющим кодом – рисунок 1.

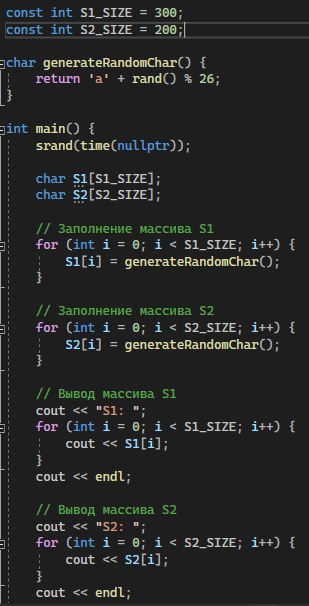


Рисунок 1 – Код для генерации массива случайных символов

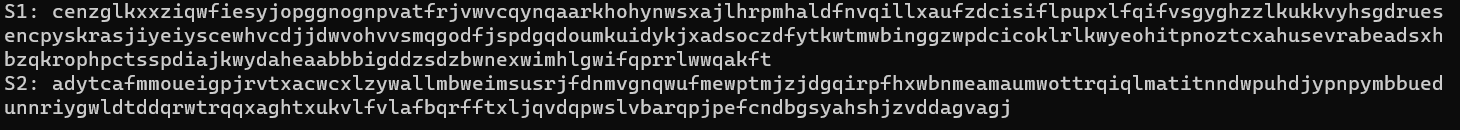


Рисунок 2 – Результат генерации

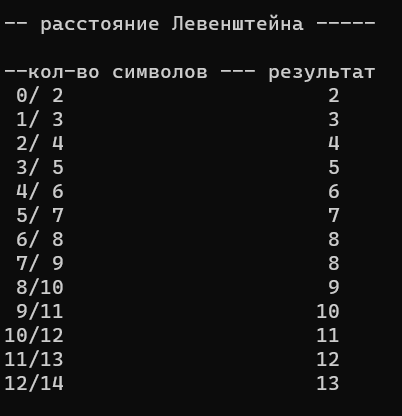


Рисунок 3 – Рассотяние Левенштейна

На рисунке 3 представлен результат расчётов дистанции Левенштейна для двух строк с максимальными длинами 12 и 15.

**Задание №2)** Вычисление дистанции Левенштейна

Будем задавать макс. длину строки – в параметрах цикла.

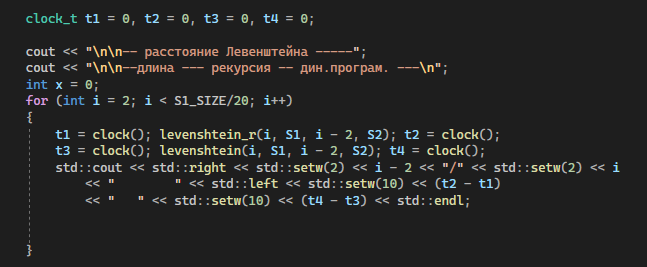


Рисунок 3 – Код основной программы

**Задание №3)** Анализ времени

Как мы можем видеть из рисунка 4, рекурсивный алгоритм работает быстро только при малом количестве символов в строке – рисунок 4.

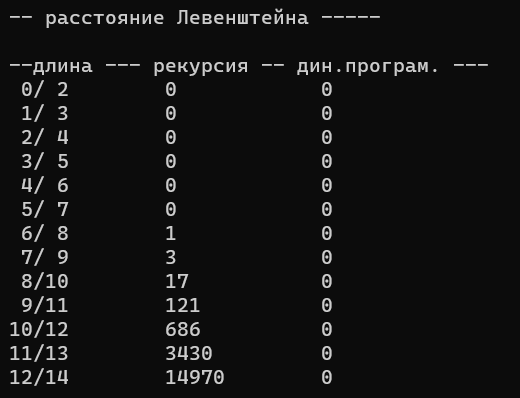


Рисунок 4 – Результат рекурсивного алгоритма

Для определения времени выполнения с помощью времени динамического программирования увеличим кол-во символов – рисунок 5.

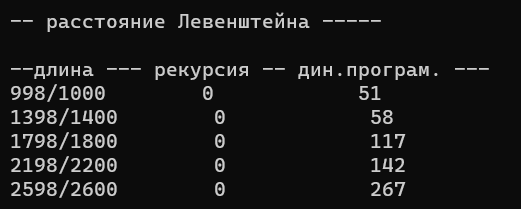


Рисунок 5 – Результат динамического программирования

Построим по полученным данным графики – рисунок 6,7.

****

Рисунок 6 – График зависимости для рекурсивного алгоритма

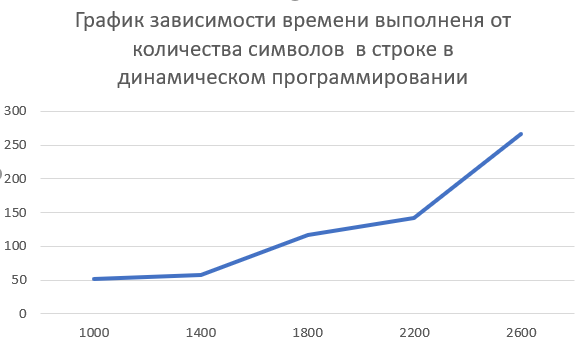


Рисунок 7 – График зависимости динамического программирования

Как мы можем видеть, динамическое программирование выигрывает как по времени, затраченному на выполнение, так и значительно превосходит количество обрабатываемых символов в строке.

**Задание №4)** Вычисление дистанции Левенштейна вручную

Для варианта 10 условие следующее - 

1. L(“ель”, “дрель”) = min
2. L(“ел”, “дрель”) = min
3. L(“ель”, “дрел”) = min
4. L(“ел”, “дрел”) = min
5. = min

L(“ ”, ”дрель”) = 5, L(“”, “дрел”) = 4

1. L(“e”, ”дрел”) =min

L(“ ”, “дрел”) = 4, L(“ ”,” дре”) = 3

1. L(“ель”, ”дре”) = min
2. = min
3. L=(“ель”, ”др”) = min
4. L=(“ель”, ”д”) = min

L=(“ель”, ””) = 3, L=(“ел”, ””) = 2

1. L=(“е”, ”дре”) = min

L=(“”, ”дре”) = 3, L=(“”, ”др”) = 2

1. L=(“ел”, ”др”) = min

L=(“е”, ”д”) = 1

1. L=(“е”, ”др”) = min

L(“”, “др”) = 2, L(“е”, “д”) = 1, L(“”, “д”) = 1

1. L=(“ел”, ”д”) = min

L(“е”, “д”) = 1, L(“ел”, “”) = 2, L(“е”, “”) = 1

1. L=(“ел”, ”д”) = min(2,3,2) = 2
2. L=(“е”, ”др”) = min(3,2,2) = 2
3. L=(“ел”, ”др”) = min(2,2,2) = 2
4. L=(“е”, ”дре”) = min(4,2,3) = 2
5. L=(“ель”, ”д”) = min(3,4,3) = 3
6. L=(“ель”, ”др”) = min(3,4,3) = 3
7. = min(3,3,3) = 3
8. L(“ель”, ”дре”) = min(4,4,3) = 3
9. L(“e”, ”дрел”) =min(5,3,4) = 3
10. = min(6,4,5) = 4
11. L(“ел”, “дрел”) = min(4,4,2) = 2
12. L(“ель”, “дрел”) = min(3,5,4) = 3
13. L(“ел”, “дрель”) = min(5,3,4) = 3
14. L(“ель”, “дрель”) = min(4,4,2) = 2

**Задание №5)** Оптимальная расстановка скобок при умножении нескольких матриц

Условие выглядит следующим образом:

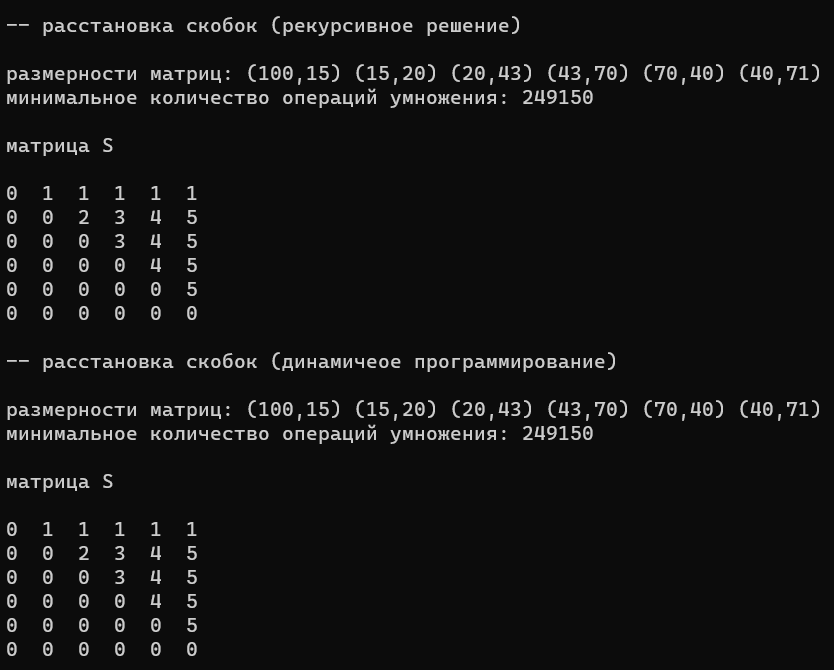


Рисунок 8 – Результат выполнения

Ячейка матрицы показывает, после какого элемента стоит скобка между матрицей с номером столбца и матрицей с номером строки. В соответствии с этим уравнение будет выглядеть следующим образом:

А1 \* ((((А2 \* А3) \* А4) \* А5) \* А6)

15\*20\*43 + 15\*43\*70+15\*70\*40+15\*40\*71+15\*71\*100 = 249150 – ответ сошёлся.

**Вывод:**

В ходе лабораторной работы были освоены общие принципы решения задач методом динамического программирования, сравнены полученные решения задач с рекурсивным методом. Получен опыт написания кода для решения задачи о дистанции Леванштейна, расстановке скобок.