МАТЕМАТИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

ЛАБОРАТОРНАЯ РАБОТА №4 – ДИСТАНЦИЯ ЛЕВЕНШТЕЙНА

ВАРИАНТ №1

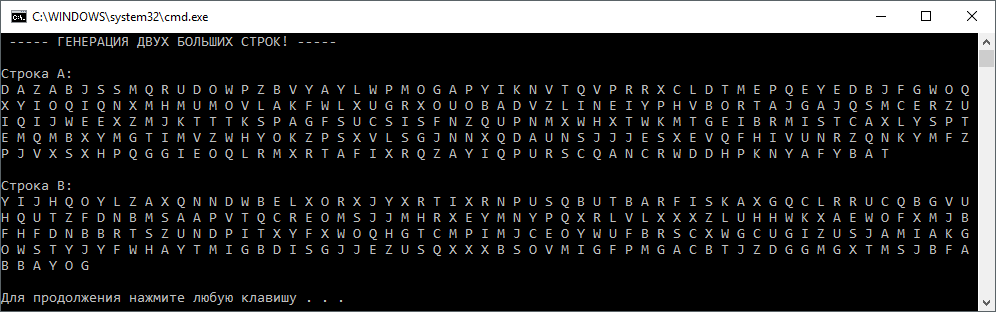
Выполнил: Демещик А.С., ПОИТ 2-3

Проверил: Блинова Е.

Минск, 2016 г.

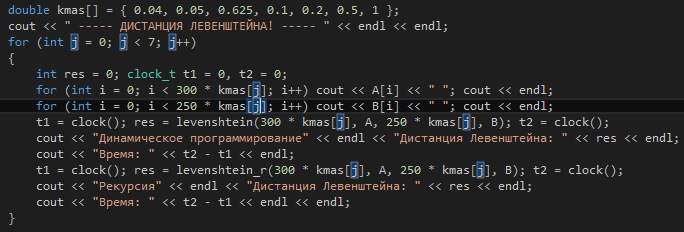
**ЗАДАНИЕ 1**

Сгенерировать две строки размерами 300 и 250 символов:

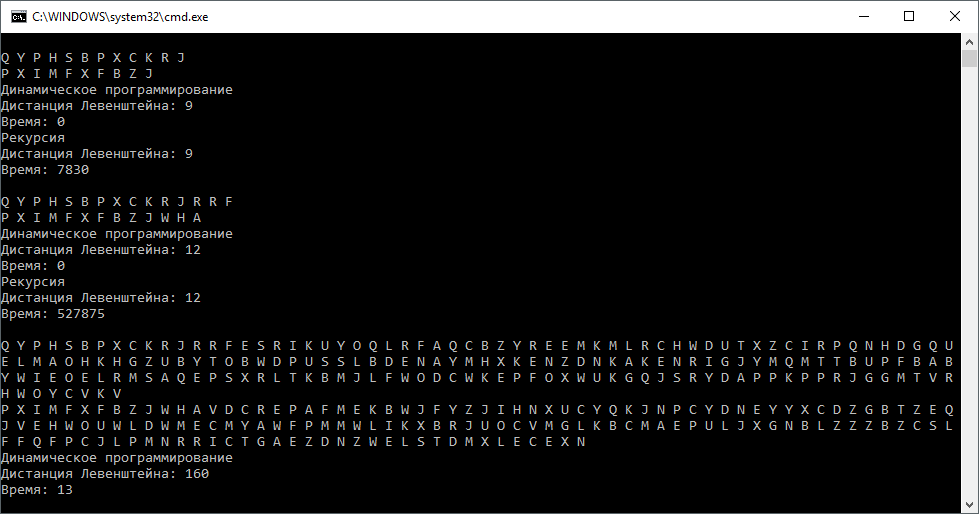


**ЗАДАНИЕ 2**

На каждом шаге меняется k (0.04, 0.05, 0.0625, 0.1, 0.2, 0.5, 1):



Пример выполнения:



**ЗАДАНИЕ 3**

Как видно из скриншота задания номер 2 и графика, представленного ниже, при самых небольших значениях k, а соответственно, при небольшой длине строк, метод динамического программирования является выигрышным вариантом по сравнению с методом рекурсии, который проигрывает. Это происходит по той причине, что в методе ДП мы должны рассмотреть полиноминальное количество вариантов, пока не найдем решение, а в методе рекурсии перебор является экспоненциальным. При кол-ве букв 160 и 120 рекурсивный метод проработал всю ночь, но так и не выдал решения.

**ЗАДАНИЕ 4**

Найти расстояние Левенштейна между словами «БЕС» и «БЛЕСК».

1. L(«БЕС», «БЛЕСК») = min

2. L(«БЕ», «БЛЕСК») = min

3. L(«БЕС», «БЛЕС») = min

4. L(«БЕ», «БЛЕС») = min

5. L(«Б», «БЛЕСК») = min

L(«», «БЛЕСК») = 5, L(«», «БЛЕС») = 4

6. L(«Б», «БЛЕС») = min

L(«», «БЛЕС») = 4, L(«», «БЛЕ») = 3

7. L(«БЕС», «БЛЕ») = min

8. L(«БЕ», «БЛЕ») = min

9. L(«Б», «БЛЕ») = min

L(«», «БЛЕ») = 3, L(«», «БЛ») = 2

10. L(«БЕС», «БЛ») = min

11. L(«БЕ», «БЛ») = min

L(«Б», «Б») = 0

12. L(«Б», «БЛ») = min

L(«», «БЛ») = 2, L(«Б», «Б») = 0, L(«», «Б») = 1

L(«Б», «БЛ») = min (3, 1, 2) = 1

13. L(«БЕС», «Б») = min

L(«БЕС», «») = 3, L(«БЕ», «») = 2

14. L(«БЕ», «Б») = min

L(«Б», «Б») = 0, L(«БЕ», «») = 2, L(«Б», «») = 1

L(«БЕ», «Б») = min (1, 3, 2) = 1

15. L(«БЕС», «Б») = min (2, 4, 3) = 2

16. L(«Б», «БЛЕ») = min (4, 2, 3) = 2

17. L(«БЕ», «БЛ») = min (2, 2, 1) = 1

18. L(«БЕС», «БЛ») = min (2, 3, 2) = 2

19. L(«БЕ», «БЛЕ») = min (3, 2, 1) = 1

20. L(«БЕС», «БЛЕ») = min (2, 3, 2) = 2

21. L(«Б», «БЛЕС») = min (5, 3, 4) = 3

22. L(«Б», «БЛЕСК») = min (6, 4, 5) = 4

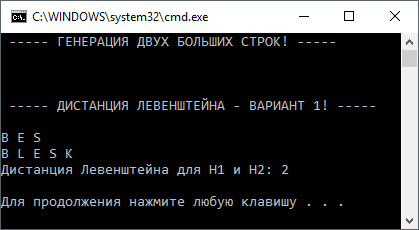
23. L(«БЕ», «БЛЕС») = min (4, 2 , 3) = 2

24. L(«БЕС», «БЛЕС») = min (3, 3, 1) = 1

25. L(«БЕ», «БЛЕСК») = min (5, 3, 4) = 3

26. L(«БЕС», «БЛЕСК») = min (4, 2, 3) = 2

Дистанция Левенштейна для слов «БЕС» и «БЛЕСК»: 2.



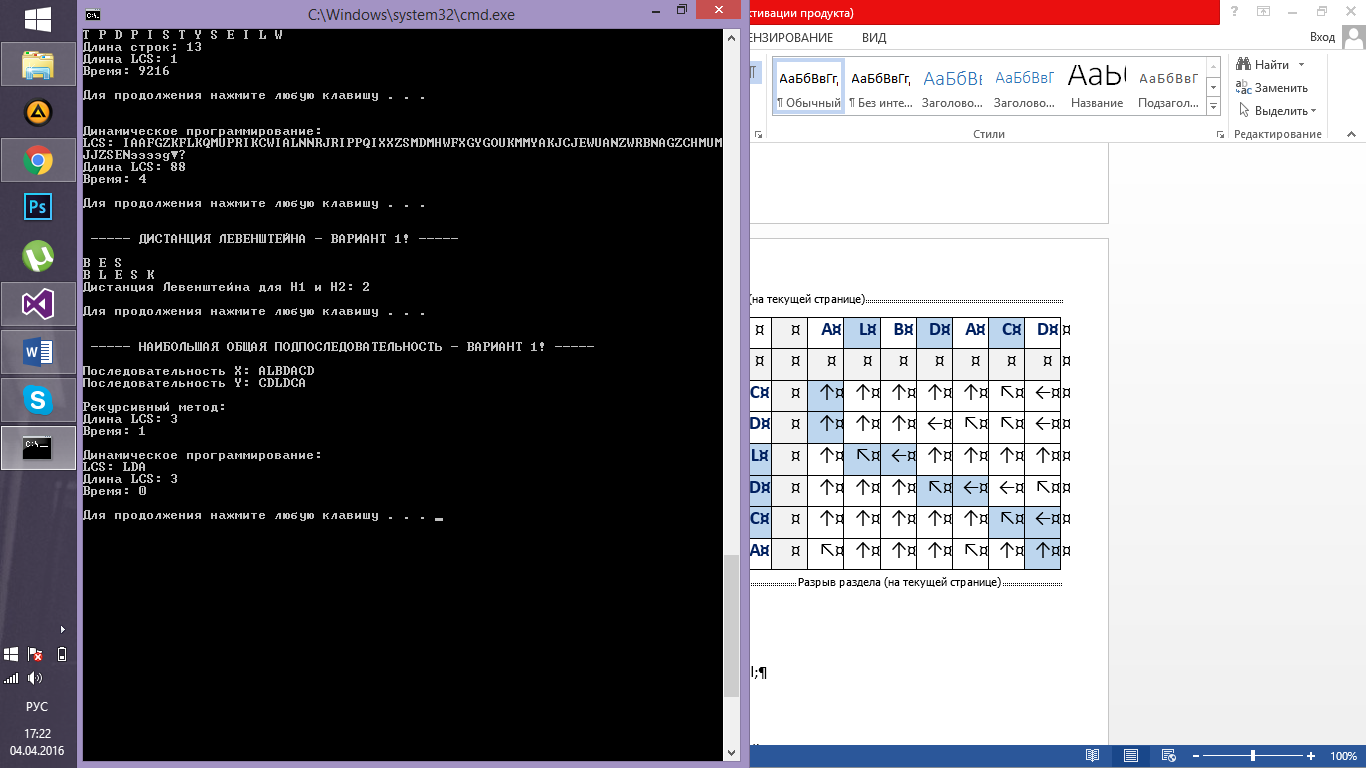
**ЗАДАНИЕ 5**

Даны две последовательности: ALBDACD и CDLDCA.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **A** | **L** | **B** | **D** | **A** | **C** | **D** |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **C** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| **D** | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 2 |
| **L** | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 2 |
| **D** | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 2 |
| **C** | 0 | 0 | 1 | 1 | 2 | 2 | 3 | 3 |
| **A** | 0 | 1 | 1 | 1 | 2 | 3 | 3 | 3 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **A** | **L** | **B** | **D** | **A** | **C** | **D** |
|  |  |  |  |  |  |  |  |  |
| **C** |  | ↑ | ↑ | ↑ | ↑ | ↑ | ↖ | ← |
| **D** |  | ↑ | ↑ | ↑ | ← | ↖ | ↖ | ← |
| **L** |  | ↑ | ↖ | ← | ↑ | ↑ | ↑ | ↑ |
| **D** |  | ↑ | ↑ | ↑ | ↖ | ← | ← | ↖ |
| **C** |  | ↑ | ↑ | ↑ | ↑ | ↑ | ↖ | ← |
| **A** |  | ↖ | ↑ | ↑ | ↑ | ↖ | ↑ | ↑ |

Проверка рекурсивным и динамичным алгоритмами:



Исследуем зависимость времени работы алгоритма от количества символов в строках:



Метод динамического программирования справляется со строками длиной 300 и 250 символов за 3 единицы тактового времени, в то время как рекурсивный метод уже на 14 символах считает почти 7 секунд (7357 единиц тактового времени). Таким образом, можно сделать вывод, что для решения задачи о нахождении наибольшей обшей подподследовательности динамическое программирование будет более эффективным методом, чем рекурсивный метод.

