1

Наибольшая общая подпоследовательность

2

Наибольшая общая подпоследов.-ность с восстановлением ответа

3

Наибольшая возрастающая подпоcледовательность

4

Наибольшая возрастающая подпоследовательность за nlogn

5

НВП за nlogn с восстановлением ответа

6

Расстояние по Левенштейну

7

НОВП

Задача 1.

Вводная

**Наибольшая общая подпоследовательность**

**Ограничение по времени работы программы: 5 секунд**

Даны две последовательности, требуется найти длину их наибольшей общей подпоследовательности.

ВХОДНЫЕ ДАННЫЕ

В первой строке входных данных содержится число N – длина первой последовательности (1⩽N⩽1000). Во второй строке заданы члены первой последовательности (через пробел) – целые числа, не превосходящие 10000 по модулю.

В третьей строке записано число M – длина второй последовательности (1⩽M⩽1000). В четвертой строке задаются члены второй последовательности (через пробел) – целые числа, не превосходящие 10000 по модулю.

ВЫХОДНЫЕ ДАННЫЕ

Требуется вывести одно число – длину наибольшей общей подпоследовательности двух данных последовательностей или 0, если такой подпоследовательности нет.

ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 3 1 2 3 4 2 3 1 5 | 2 |

Начало формы

Задача решена верно

OK. Your score is = 100, 20/20 tests passed

Решение задачи

Будем использовать динамическое программирование. Целевая функция - F(i,j) - длина наибольшей общей подпоследовательности для префиксов A[:i] и B[:j]. Более подробно решение задачи изложено в теоретическом материале.

Конец формы

Задача 2.

Лёгкая

**Наибольшая общая подпоследов.-ность с восстановлением ответа**

**Ограничение по времени работы программы: 5 секунд**

Даны две последовательности, требуется найти и вывести их наибольшую общую подпоследовательность.

ВХОДНЫЕ ДАННЫЕ

В первой строке входных данных содержится число N – длина первой последовательности (1⩽N⩽1000). Во второй строке заданы члены первой последовательности (через пробел) – целые числа, не превосходящие 10000 по модулю.

В третьей строке записано число M – длина второй последовательности (1⩽M⩽1000). В четвертой строке задаются члены второй последовательности (через пробел) – целые числа, не превосходящие 10000 по модулю.

ВЫХОДНЫЕ ДАННЫЕ

Требуется вывести наибольшую общую подпоследовательность данных последовательностей, через пробел. Если таких последовательностей несколько, выведите любую из них.

ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 3 1 2 3 4 2 3 1 5 | 2 3 |

Начало формы

Задача решена верно

OK. Your score is = 100, 20/20 tests passed

Решение задачи

Сначала при помощи динамического программирования найдем длину наибольшей общей подпоследовательности, потом восстановим ответ при помощи «обратного прохода». Более подробно решение задачи изложено в теоретическом материале.

Конец формы

Задача 3.

Лёгкая

**Наибольшая возрастающая подпоcледовательность**

**Ограничение по времени работы программы: 2 секунды**

Дана последовательность целых чисел. Постройте наибольшую возрастающую подпоследовательность данной последовательности.

ВХОДНЫЕ ДАННЫЕ

В первой строке входных данных записано число элементов последовательности N, 1⩽N⩽1000. Во второй строке записаны N целых чисел через пробел.

ВЫХОДНЫЕ ДАННЫЕ

Требуется вывести наибольшую возрастающую подпоследовательность данной последовательности (последовательность чисел через пробел). Если таких подпоследовательностей несколько, необходимо вывести одну (любую) из них.

ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 6 3 29 5 5 28 6 | 3 5 28 |

Начало формы

Задача решена верно

OK. Your score is = 100, 18/18 tests passed

Решение задачи

Сначала при помощи динамического программирования найдем длину наибольшей возрастающей подпоследовательности, потом восстановим ответ при помощи «обратного прохода». Более подробно решение задачи изложено в теоретическом материале.

Конец формы

Задача 4.

Средняя

**Наибольшая возрастающая подпоследовательность за nlogn**

Числовая последовательность задана рекуррентной формулой:  ai+1=(k∗ai+b)mod m. Найдите длину её наибольшей возрастающей подпоследовательности.

ВХОДНЫЕ ДАННЫЕ

Программа получает на вход пять целых чисел: длину последовательности  n (1≤n≤105), начальный элемент последовательности a1, параметры k,b,m для вычисления последующих членов последовательности (1≤m≤104,0≤k<m,0≤b<m,0≤a1<m).

ВЫХОДНЫЕ ДАННЫЕ

Требуется вывести длину наибольшей возрастающей подпоследовательности данной последовательности.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 5 41 2 1 100 | 3 |

Задача 5.

Сложная

**НВП за nlogn с восстановлением ответа**

Числовая последовательность задана рекуррентной формулой:  ai+1=(k∗ai+b)mod m. Найдите длину её наибольшей возрастающей подпоследовательности.

ВХОДНЫЕ ДАННЫЕ

Программа получает на вход пять целых чисел: длину последовательности  n (1≤n≤105), начальный элемент последовательности a1, параметры k,b,m для вычисления последующих членов последовательности (1≤m≤104,0≤k<m,0≤b<m,0≤a1<m).

ВЫХОДНЫЕ ДАННЫЕ

Требуется вывести наибольшую возрастающую подпоследовательность данной последовательности, разделяя числа пробелами. Если таких последовательностей несколько, необходимо вывести одну (любую) из них.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 5 41 2 1 100 | 41 67 71 |

Задача 6.

Олимпиадная

**Расстояние по Левенштейну**

Дана текстовая строка. С ней можно выполнять следующие операции: 1. Заменить один символ строки на другой символ. 2. Удалить один произвольный символ. 3. Вставить произвольный символ в произвольное место строки. Например, при помощи первой операции из строки "СОК" можно получить строку "СУК", при помощи второй операции - строку "ОК", при помощи третьей операции - строку "СТОК. Минимальное количество таких операций, при помощи которых можно из одной строки получить другую, называется стоимостью редактирования или расстоянием Левенштейна. Определите расстояние Левенштейна для двух данных строк.

ВХОДНЫЕ ДАННЫЕ

Программа получает на вход две строки, длина каждой из которых не превосходит 1000 символов, строки состоят только из заглавных латинских букв.

ВЫХОДНЫЕ ДАННЫЕ

Требуется вывести одно число – расстояние Левенштейна для данных строк.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| ABCDEFGH ACDEXGIH | 3 |

Задача 7.

Сложная

**НОВП**

Даны две последовательности. Найдите их наибольшую общую возрастающую подпоследовательность

ВХОДНЫЕ ДАННЫЕ

Во входном файле записаны две последовательности. Каждая последовательность описывается двумя строками следующим образом: в первой строке идет длина последовательности  M (1≤M≤500), во второй идут M целых чисел  ai (−231≤ai<231) – члены последовательности.

ВЫХОДНЫЕ ДАННЫЕ

В первой строке выходного файла выведите N - длину наибольшей общей возрастающей подпоследовательности. Во второй строке выходного файла выведите саму подпоследовательность.

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 5 1 4 2 5 -12 4 -12 1 2 4 | 2  1 2 |