СЛОЖНАЯ ЗАДАЧА

УСЛОВИЕ

№17 ИЗ ЕГЭ

РЕШЕНИЕ

Дан файл 17(2).txt с последовательность целых чисел, нужно найти самую длинную возрастающую подпоследовательность (LIS) и вторую по длине возрастающую подпоследовательность (2LIS). В ответе указать их длинны по порядку: LIS и 2LIS. Если существует несколько классов одинаковой длины, вернуть любую из них.

ШАГИ РЕШЕНИЯ:

1. Определим функцию longest\_increasing\_subsequence, которая принимает список целых чисел в качестве входных данных.

2. Инициализируем два массива, lis и second\_lis, с той же длиной, что и входной список, и заполните их значением 1. Эти массивы будут хранить длины LIS и 2LIS, заканчивающиеся на каждом индексе.

3. Выполним итерацию по входному списку, начиная со второго элемента. Для каждого элемента сравним его со всеми предыдущими элементами. Если текущий элемент больше предыдущего элемента и длина LIS, заканчивающегося на предыдущем элементе плюс 1, больше длины LIS, заканчивающейся на текущем элементе, обновим длину LIS, заканчивающейся на текущем элементе.

4. После перебора всего списка найдем максимальную длину LIS и сохраним его индекс.

5. Выполним повторную итерацию по входному списку, начиная с индекса максимальной длины LIS. Для каждого элемента сравним его со всеми предыдущими элементами. Если текущий элемент больше предыдущего элемента и длина 2LIS, заканчивающегося на предыдущем элементе плюс 1, больше длины 2LIS, заканчивающегося на текущем элементе, а длина LIS, заканчивающегося на предыдущем элементе плюс 1, меньше максимальной длины LIS, обновите LIS. длина 2LIS, заканчивающегося на текущем элементе.

6. После перебора всего списка найдем максимальную длину 2LIS и сохраним его индекс.

7. Инициализируем два пустых списка, lis\_subsequence и second\_lis\_subsequence, для хранения подпоследовательностей LIS и 2LIS.

8. Отследим LIS и 2LIS, используя массивы lis и second\_lis, начиная с индексов, найденных на шагах 4 и 6. Добавим элементы LIS и 2LIS в соответствующие списки подпоследовательностей, уменьшая при этом максимальную длину.

9. Перевернем списки подпоследовательностей, чтобы получить правильный порядок.

10. В качестве результата возвращем подпоследовательности LIS и 2LIS.

11. В основной части кода считаем список целых чисел из файла с помощью функции open() и преобразуем содержимое в целые числа.

12. Вызовим функцию longest\_increasing\_subsequence() со списком целых чисел в качестве входных данных и сохраним подпоследовательности LIS и 2LIS.

13. Выведим длины подпоследовательностей LIS и 2LIS.

КОД:

def longest\_increasing\_subsequence(nums):

n = len(nums)

lis = [1] \* n

for i in range(1, n):

for j in range(i):

if nums[i] > nums[j] and lis[i] < lis[j] + 1:

lis[i] = lis[j] + 1

max\_length = max(lis)

max\_index = lis.index(max\_length)

second\_lis = [1] \* n

for i in range(1, n):

for j in range(i):

if nums[i] > nums[j] and second\_lis[i] < second\_lis[j] + 1 and lis[j] + 1 < max\_length:

second\_lis[i] = second\_lis[j] + 1

second\_max\_length = max(second\_lis)

second\_max\_index = second\_lis.index(second\_max\_length)

lis\_subsequence = []

second\_lis\_subsequence = []

i = max\_index

while i >= 0 and max\_length > 0:

if lis[i] == max\_length:

lis\_subsequence.append(nums[i])

max\_length -= 1

i -= 1

lis\_subsequence.reverse()

i = second\_max\_index

while i >= 0 and second\_max\_length > 0:

if second\_lis[i] == second\_max\_length:

second\_lis\_subsequence.append(nums[i])

second\_max\_length -= 1

i -= 1

second\_lis\_subsequence.reverse()

return lis\_subsequence, second\_lis\_subsequence

with open('17(2).txt', "r") as file:

nums = [int(num) for num in file.read().split()]

lis, second\_lis = longest\_increasing\_subsequence(nums)

print("LIS:", len(lis))

print("2LIS:", len(second\_lis))

ОТВЕТ ДЛЯ ДАННОГО ФАЙЛА: 6 5