1. Найдите подпоследовательность заданной последовательности такую, чтобы сумма подпоследовательности была как можно больше, а элементы подпоследовательности отсортированы в возрастающем порядке. Эта подпоследовательность не обязательно непрерывна или уникальна.

Пример: {0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11} => {8, 12, 14}.

1. Найти подпоследовательность заданной последовательности, в которой элементы подпоследовательности отсортированы в порядке от низшего к высшему, и в которой подпоследовательность является максимально возможной. Эта подпоследовательность не обязательно непрерывна или уникальна.

Пример: [0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15] -> [0, 2, 6, 9 , 11, 15]. (Самая длинная возрастающая подпоследовательность в этом примере не уникальна, например, [0, 4, 6, 9, 11, 15] или [0, 4, 6, 9, 13, 15])

1. Найти количество n-значных двоичных чисел, не содержащих двух подряд идущих единиц.

Пример: n=5 -> 13 (00000, 00001, 00010, 00100, 00101, 01000, 01001, 01010, 10000, 10001, 10010, 10100, 10101)

1. Найти наименьшую общую суперпоследовательность, то есть такую последовательность, что обе последовательности являются ее подпоследовательностями (не обязательно подстрокой!)

Пример: ABCBDAB и BDCABA -> ABCBDCABA, ABDCABDAB или ABDCBDABA.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | \_ | A | B | C | B | D | A | B |
| \_ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| B | 1 | 2 | 2 | 3 | 4 | 5 | 6 | 7 |
| D | 2 | 3 | 3 | 4 | 5 | 5 | 6 | 7 |
| C | 3 | 4 | 4 | 4 | 5 | 6 | 7 | 8 |
| A | 4 | 4 | 5 | 5 | 6 | 7 | 7 | 8 |
| B | 5 | 5 | 5 | 6 | 6 | 7 | 8 | 8 |
| A | 6 | 6 | 6 | 7 | 7 | 8 | 8 | 9 |

F(S\_n, T\_m) =

Если s[n] == t[m]:



F(S\_n, T\_m) = F(S\_n-1, T\_m-1) + 1



Иначе

F(S\_n, T\_m) = min(F(S\_n, T\_m-1), F(S\_n-1, T\_m)) + 1

1. Найти наибольшую общую подстроку.

Пример: ABABC и BABCA -> BABC.



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | \_ | A | B | A | B | C |
| \_ | 0 | 0 | 0 | 0 | 0 | 0 |
| B | 0 | 0 | 1 | 1 | 1 | 1 |
| A | 0 | 1 | 1 | 2 | 2 | 2 |
| B | 0 | 1 | 2 | 2 | 3 | 3 |
| C | 0 | 1 | 2 | 2 | 3 | 4 |
| A | 0 | 1 | 2 | 3 | 3 | 4 |

F(S\_n,T\_m) =

Если s[n]==t[m]:

F(S\_n,T\_m) = F(S\_n-1,T\_m-1) + 1

Иначе

F(S\_n,T\_m) = max(F(S\_n,T\_m-1), F(S\_n-1,T\_m))

F(ABABC, BABCA) = max(F(ABAB, BABCA), F(ABABC, BABC))

F(ABABC, BABC) = F(ABAB, BAB) + 1

F(ABAB, BAB) = F(ABA, BA) + 1

1. Найти самую длинную повторяющуюся подпоследовательность (не обязательно подстроку!)

Пример: ATACTCGGA -> ATCG

1. Найти для данной строки длиннейшую подпоследовательность, которая является палиндромом

Пример: ABBDACAB -> BACAB

1. Дан набор чисел. Необходимо найти длину наибольшей чередующейся последовательности, то есть последовательности, в которой a1<a2>a3<a4… или аналогично начиная со знака >

Пример: [8, 9, 6, 4, 5, 7, 3, 2, 4] -> 6 (например, [8, 9, 4, 7, 2, 4])

1. Дана строка и дан набор слов. Необходимо определить, может ли строка быть разбита на слова, содержащиеся в словаре.

Пример:

d = ['self', 'th', 'is', 'famous', 'Word', 'break', 'b', 'r', 'e', 'a', 'k', 'br', 'bre', 'brea', 'ak', 'problem']

s = ‘Wordbreakproblem’

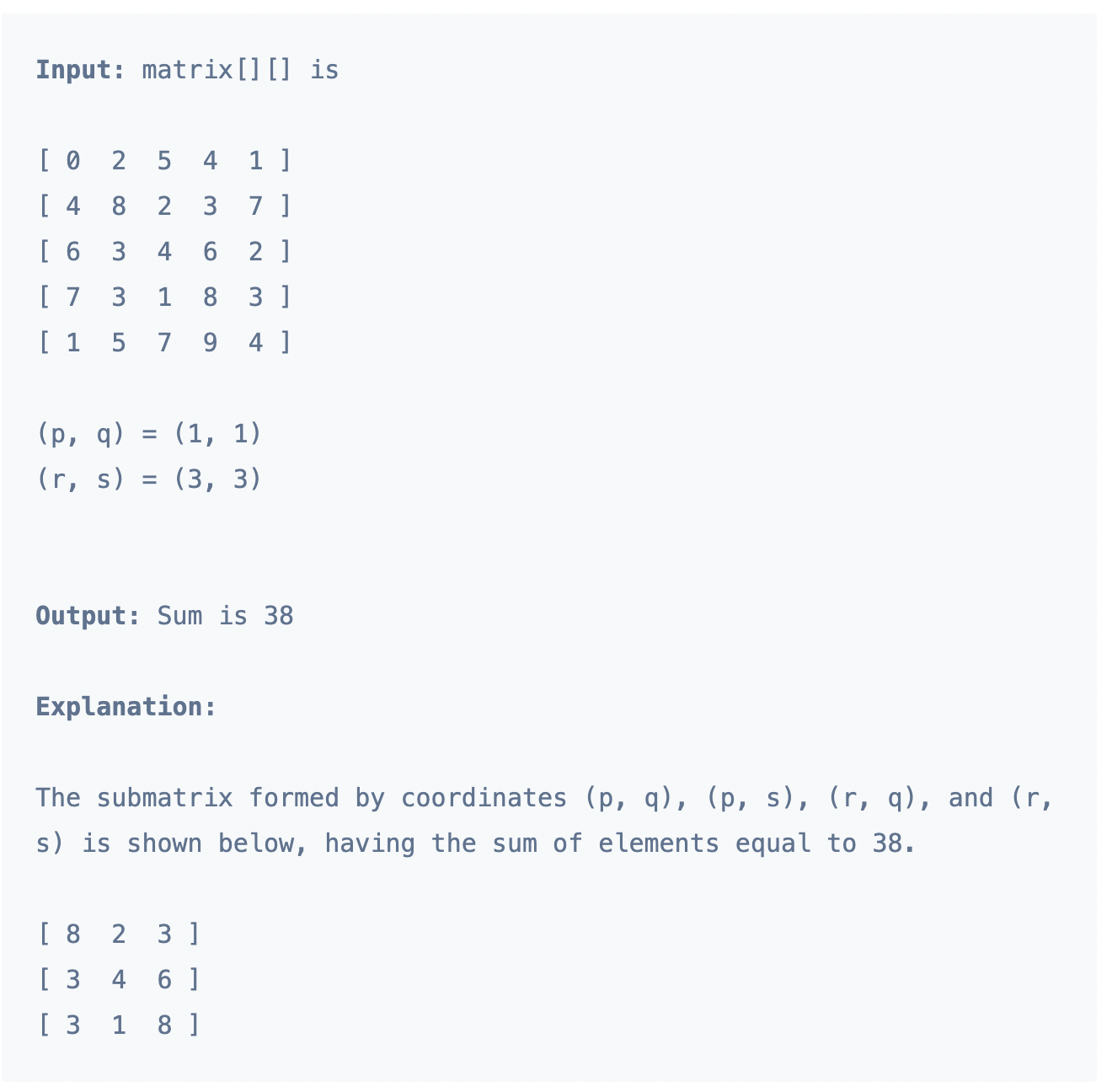
Вывод: True (Word break problem или Word brea k problem или Word br e ak problem или другие)

1. Дана матрица MxN, заполненная целыми числами - стоимостью ячейки. Надо найти самый дешевый путь из верхнего левого угла (0, 0) в нижний правый (M-1, N-1). Перемещаться можно только вправо и вниз. В примере ниже путь выделен синим, а его стоимость равна 36.

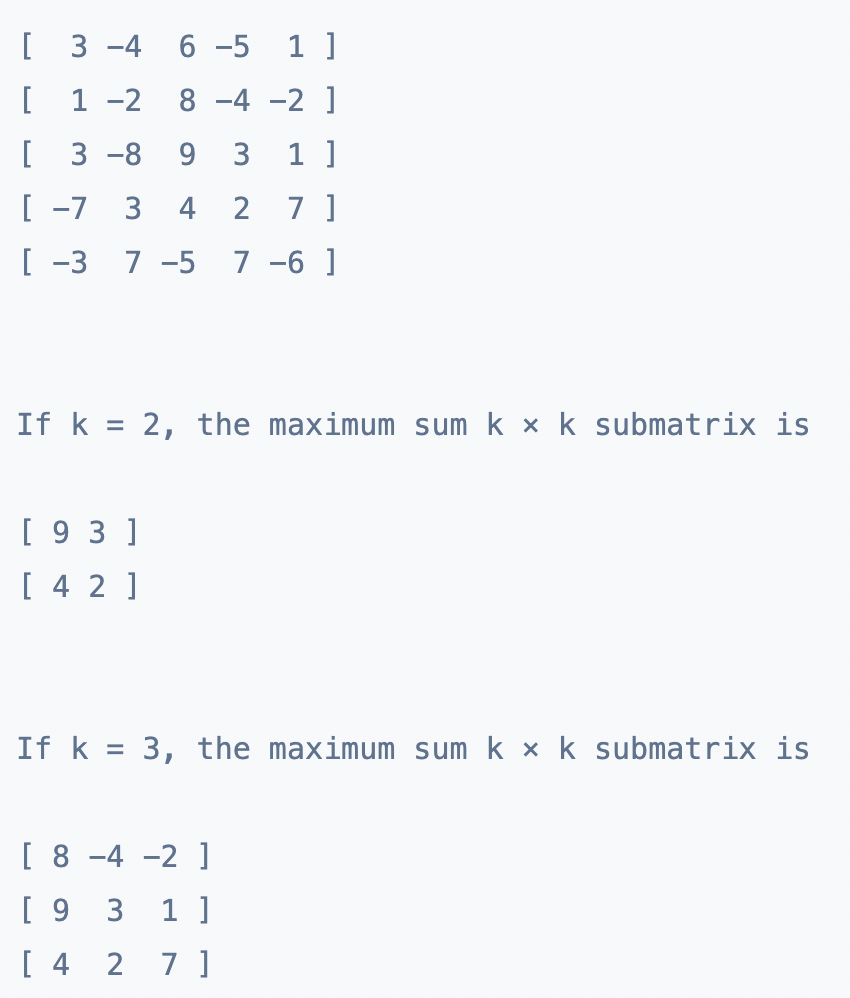
**A picture containing diagram

Description automatically generated**

1. Дана матрица MxN. Для координат (p, q) и (r, s), обозначающих верхний левый и правый нижний угол подматрицы, необходимо вернуть сумму элементов этой подматрицы.



1. Дана матрица MxN. Необходимо вернуть максимальную сумму для подматрицы размера KxK.



1. Дана матрица MxN, заполненная нулями и единицами. Найти размер наибольшей матрицы, состоящей только из единиц. Для примера ниже ответ 3.

**Table

Description automatically generated**