

Для каждого элемента из первой строки производится поиск элемента из второй строки с наименьшим индексом, образующего с ним пару. Если пара найдена, то функция поиска вызывает себя, с обновлёнными (“обрезанными” по индексы элементов, из которых составлена пара) первой и второй строками. Функция поиска возвращает максимальное количество найденных пар при выборе любой возможной пары. Оптимизация:

1. Поиск не производится для элементов из первой строки, равных предыдущему элементу т.к. они образуют пару с одним элементом из второй строки, но при этом уменьшают количество элементов из первой строки для последующих пар.
2. Для каждого элемента из второй строки хранится наибольшее количество пар образованных с использованием элементов с меньшими индексами. Если пара найдена, но количество полученных ранее пар в этой последовательности не превосходит текущее значение для элемента пары из второй строки, то последовательность не рассматривается, иначе значение в массиве обновляется.
3. Для поиска второго элемента пары используется алгоритм бинарного поиска. Для его применения элементы из второй строки записываются в массив структур вместе с индексами (индексы уникальны и соответствуют порядку ввода элементов). Далее массив сортируется по возрастанию значений элементов (для элементов с одинаковыми значениями по возрастанию индексов). Функция бинарного поиска возвращает наименьший индекс элемента из элементов, с индексом, превосходящим нижнюю границу или константу (NotFound), в случае если такого элемента нет.

Асимптотическая оценка времени работы алгоритма:

Бинарного поиска - $\log_2 n$

Перебора элементов первой строки – n

Рекурсивный вызов функции поиска (с учётом оптимизации) - n

Сортировки - n^2

Итого - $n^2 + n^2 \log_2 n \sim n^2 \log_2 n$

Асимптотическая оценка памяти, потребляемой алгоритмом:

Первая строка – n

Вторая строка (с индексами) – $2n$

Массив количества полученных ранее пар – n

Итого – $4n$