Для нахождения процента совпадения строк или текстов используются различные решения.

1. Метрика [Левенштейна](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D0%B5_%D0%9B%D0%B5%D0%B2%D0%B5%D0%BD%D1%88%D1%82%D0%B5%D0%B9%D0%BD%D0%B0)

В данном решении используется 3 библиотеки.

* Fuzz
* Levenshtein
* statistics

**Расстояние Левенштейна** (*редакционное расстояние*, *дистанция редактирования*) — [метрика](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D1%80%D0%B8%D0%BA%D0%B0_(%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%B3%D0%B5%D0%BE%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%8F)), измеряющая по модулю разность между двумя последовательностями символов. Она определяется как минимальное количество односимвольных операций (а именно вставки, удаления, замены), необходимых для превращения одной последовательности символов в другую. В общем случае, операциям, используемым в этом преобразовании, можно назначить разные цены. Широко используется в [теории информации](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%B8) и [компьютерной лингвистике](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BB%D0%B8%D0%BD%D0%B3%D0%B2%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0).

1. Библиотека difflib

Это гибкий класс для сравнения пар последовательностей любого типа, при условии, что элементы последовательности являются [хешируемыми](https://docs.python.org/3/glossary.html#term-hashable).

Объект является хешируемым, если он имеет хеш-значение, которое никогда не меняется в течение его жизни (для этого нужен \_\_hash\_\_()метод), и его можно сравнить с другими объектами (для этого нужен \_\_eq\_\_()метод). Хэшируемые объекты, которые сравниваются равными, должны иметь одинаковое хэш-значение.

Хешируемость делает объект пригодным для использования в качестве ключа словаря и элемента набора, поскольку эти структуры данных используют значение хэша внутри.

Идея состоит в том, чтобы найти самую длинную непрерывную совпадающую подпоследовательность, которая не содержит “ненужных” элементов; эти “ненужные” элементы в некотором смысле неинтересны, например, пустые строки или пробелы. Затем та же идея рекурсивно применяется к фрагментам последовательностей слева и справа от соответствующей подпоследовательности. Это не дает минимальных последовательностей редактирования, но, как правило, дает совпадения, которые “выглядят правильно” для людей.