- **Возврат нескольких значений из функций.** Кортежи позволяют вернуть несколько значений из функции в виде одного объекта. Например, функция может возвращать кортеж с координатами — x, y — точки.
- **Представление неизменяемых данных.** Если нужно хранить набор данных, который не должен быть изменён, можно использовать кортежи. Например, применять кортеж для представления даты и времени или информации о человеке (имя, возраст, пол и так далее).

Алгоритмическая сложность методов кортежа в Python:

- Доступ к элементу по индексу (кортеж[индекс]):
  - средний случай: O(1);
  - худший случай: O(1).
    Доступ к элементу кортежа по индексу всегда выполняется за одно и то же время, поскольку кортеж неизменяем и индекс элемента напрямую связан с его позицией в памяти.
- Поиск элемента (in):
  - средний случай: O(n);
  - худший случай: O(n).
    Чтобы найти один элемент в кортеже, в худшем случае придётся пройти по всем. В среднем случае нужно будет пройти по половине элементов, но и в этом случае у нас сохранится линейная зависимость от количества элементов в кортеже (чем больше элементов в кортеже, тем больше их и в половине кортежа).
- Вставка элемента:
  - средний случай: O(n);
  - худший случай: O(n).
    Вставка элемента требует создания нового кортежа большего размера и копирования всех элементов. А значит, у нас и тут получается линейная зависимость времени выполнения вставки элемента от количества элементов в кортеже.
- Удаление элемента:
  - средний случай: O(n);
  - худший случай: O(n).
    Удаление, как и вставка, требует создания нового кортежа без ненужного элемента и копирования всех остальных элементов. Это тоже требует линейного времени.

Кортежи обеспечивают удобное хранение неизменяемых данных, имеют простой и понятный синтаксис. Однако если понадобится изменять или расширять набор данных, то, возможно, стоит использовать списки.

Хеширование

Простота кортежей, с одной стороны, даёт преимущества (кортежи занимают мало памяти), но с другой — у неё есть и недостатки.

Например, если мы захотим проверить, есть ли какой-то объект внутри кортежа, Python пройдёт по всем объектам, сравнивая их с целевым. Соответственно, в худшем случае