|  |  |
| --- | --- |
| Str | Сложность |
| Получение элемента, s[i] | О(1) |
| Размер списка, len(s) | О(1) |
| Получение среза, s[a:b] | O(b-a) |

**Операции со строками (Str)**

Самое базовое понятие статического анализа – **O(1)**. Этот класс сложности имеют операции, которые выполняются за константное время

**Получение элемента, s[i]:**

* Строка является последовательностью с доступом к элементам по индексам. Получение элемента по индексу — операция с постоянной сложностью, так как доступ осуществляется напрямую, без перебора других элементов.

**Размер списка, len(s):**

* Длина строки хранится в виде атрибута объекта, поэтому вызов len(s) выполняется за постоянное время, так как не требуется проходить по всем символам.

**Получение среза, s[a:b]:**

* Для создания среза необходимо скопировать элементы из диапазона [a:b], что требует линейного времени относительно длины среза. Поэтому сложность зависит от числа копируемых элементов.

**Операции с множествами (Set)**

|  |  |
| --- | --- |
| Set | Сложность |
| Размер множества, len(s) | O(1) |
| Добавление элемента, s.add(x) | O(1) |
| Проверка наличия значения, x in/not in s | O(1) |
| Перебор множества, for v in s: | O(N) |
| Объединение (union), s | t | O(len(s)+len(t)) |

**Размер множества, len(s):**

* Как и в случае со строками, размер множества в Python хранится в виде атрибута объекта, и его получение выполняется за постоянное время.

**Добавление элемента, s.add(x):**

* Множества реализованы на основе хеш-таблиц. В среднем добавление элемента выполняется за постоянное время. Однако при сильных коллизиях (например, если хеши распределены неравномерно) сложность может увеличиться.

**Проверка наличия значения, x in s / x not in s:**

* Проверка принадлежности элемента множеству также основана на хешировании. В среднем это занимает O(1), но при коллизиях может потребоваться O(n).

**Перебор множества, for v in s:**

* Для перебора всех элементов множества требуется пройти по каждому элементу, поэтому сложность пропорциональна числу элементов в множестве. Поэтому N - количество элементов в множестве.

**Объединение (union), s | t:**

* Операция объединения требует добавления всех элементов второго множества t в первое множество s. Это зависит от количества элементов в обоих множествах. Поэтому если s содержит m элементов, а t содержит n элементов, то сложность будет O(m + n), что равно O(len(s) + len(t)).