**1. Ініціалізація:**

* \_\_init\_\_():
  + Розрахунок розміру масиву m та кількості хеш-функцій k займає O(1).
  + Ініціалізація масиву лічильників self.counter\_array розміром m займає O(m).
  + Генерація коефіцієнтів a та b займає O(k), оскільки для кожної хеш-функції створюються випадкові коефіцієнти.
  + Загальна складність ініціалізації: O(m+k).

**2. Обчислення розміру масиву calculate\_size():**

* Виконується за допомогою формули: Изображение выглядит как Шрифт, текст, число, Графика

  Автоматически созданное описание
  + Формула розрахунку має постійну складність O(1), оскільки виконує арифметичні операції.

**3. Обчислення кількості хеш-функцій calculate\_hash\_count():**

* Виконується за формулою: Изображение выглядит как Шрифт, символ, рукописный текст, Графика

  Автоматически созданное описание
  + Формула також має постійну складність O(1).

**4. Додавання елемента add(item)**:

* Виконує такі дії:
  + Викликає get\_hash\_values(item) для обчислення k хеш-значень.
    - get\_hash\_values(item) викликає hash\_item() k разів.
    - hash\_item() виконує операції з числом, що мають складність O(1).
    - Перетворення рядка в ціле число string\_to\_int() має складність O(L), де L — довжина рядка, оскільки кожен символ обробляється один раз.
    - Загальна складність get\_hash\_values() дорівнює O(k⋅L).
  + Після обчислення хешів виконує збільшення лічильників для кожного індексу, що займає O(k).
  + Загальна складність add(item) дорівнює O(k⋅L)+O(k)=O(k⋅L).

**5. Перевірка наявності check(item)**:

* Аналогічно до add() викликає get\_hash\_values() з тією ж складністю O(k⋅L).
* Перевірка значень лічильників займає O(k).
* Загальна складність check(item) дорівнює O(k⋅L).

**6. Видалення елемента remove(item)**:

* Спочатку виконує перевірку наявності елемента за допомогою check(item), що має складність O(k⋅L).
* Якщо елемент знайдено, зменшує значення лічильників, що займає O(k)
* Загальна складність remove(item) дорівнює O(k⋅L).

**Загальна асимптотична складність:**

* Ініціалізація: O(m+k)
* Додавання елемента: O(k⋅L)
* Перевірка наявності: O(k⋅L)
* Видалення елемента: O(k⋅L)