Алгоритмы и структуры данных-2

SET 5. Задача A2.

Весна 2024. Клычков М. Д.

Пункт 1. В текстовом анализе не будем учитывать константы для функций квадратичного и кубического пробирования, так как, несмотря на то что «правильные» константы могут обеспечить лучшую заполненность хеш-таблицы, при достаточно больших M асимптотически выбор константы не влияет на результат. Более того «оптимальных» констант достаточно много (например, для квадратичного пробирования). Объявим все константы для квадратичного и кубического пробирования равными $c_1 = c_2 = c_3 = 1$.

$$hash_2(k, i) = (hash(k) + i + i^2) \mod M$$
$$hash_3(k, i) = (hash(k) + i + i^2 + i^3) \mod M$$

Сравним два метода пробирования по скорости возникновения кластеров. Для этого вычислим расстояние между двумя последовательными ячейками для одного ключа, то есть расстояние между элементами во вторичном кластере:

$$\Delta_2(i) = hash_2(k, i+1) - hash_2(k, i) = 2 + 2i$$
$$\Delta_3(i) = hash_3(k, i+1) - hash_3(k, i) = 3 + 5i + 3i^2$$

При такой записи становится очевидно, что коллизии в кубическом пробировании хранятся на большем расстоянии, более разреженно, в отличие от квадратичного. *Когда большие расстояние между коллизиями могут быть полезны?* Например, такой подход позволяет снизить количество первичных кластеров, то есть локальные «островки» занятых ячеек в массиве.

Какие могут быть минусы у кубического пробирования? Ему свойственны все те же минусы, что есть у квадратичного пробирования по сравнению с линейным: невозможность делать шаги назад (например, при сдвиге для удавления), возможность посещения одной и той же ячейки (т.е. зацикливание), а следовательно большое количество шагов для посещения хотя бы половины хеш-таблицы. При сравнении с кубическим пробированием, озвученные проблемы только усугубляются, например, при маленьком M каждый шаг может приводить в уже посещенную ячейку с большей вероятностью.

Пункт 2. Для проверки предлагается выполнить последовательную вставку множества предварительно случайно сгенерированных элементов в хеш-таблицу размера M методами квадратичного и кубического пробирования. Будем варьировать количество ячеек M и объем выборки для вставки (по этим параметрам можно будет также получить коэффициент заполненности).

```python
def quadratic_probing(hash_table, key, M):
    i = 0 # number of probes - 1
    index = hash(key, M)
    while hash_table[index] is not None:
        i += 1
        index = (hash(key, M) + i + i**2) % M
    hash_table[index] = key
    return i + 1

def cubic_probing(hash_table, key, M):
    i = 0 # number of probes - 1
    index = hash(key, M)
    while hash_table[index] is not None:
        i += 1
        index = (hash(key, M) + i + i**2 + i**3) % M
    hash_table[index] = key
    return i + 1
```

Рис. 1: Реализация квадратичного и кубического пробирования

```python
def simulate_probing(M, num_keys):
    keys = random.sample(range(1, 10**6), num_keys)
    hash_table_quadratic = [None] * M
    hash_table_cubic = [None] * M
    total_probes_quadratic = 0
    sum_probes_quadratic = 0 # for calculating average
    total_probes_cubic = 0
    sum_probes_cubic = 0 # for calculating average

    for key in keys:
        probes = quadratic_probing(hash_table_quadratic, key, M)
        total_probes_quadratic += probes
        sum_probes_quadratic += probes

        probes = cubic_probing(hash_table_cubic, key, M)
        total_probes_cubic += probes
        sum_probes_cubic += probes
    return (total_probes_quadratic, sum_probes_quadratic/num_keys, total_probes_cubic, sum_probes_cubic/num_keys)
```

Рис. 2: Сбор статистики для квадратичного и кубического пробирования

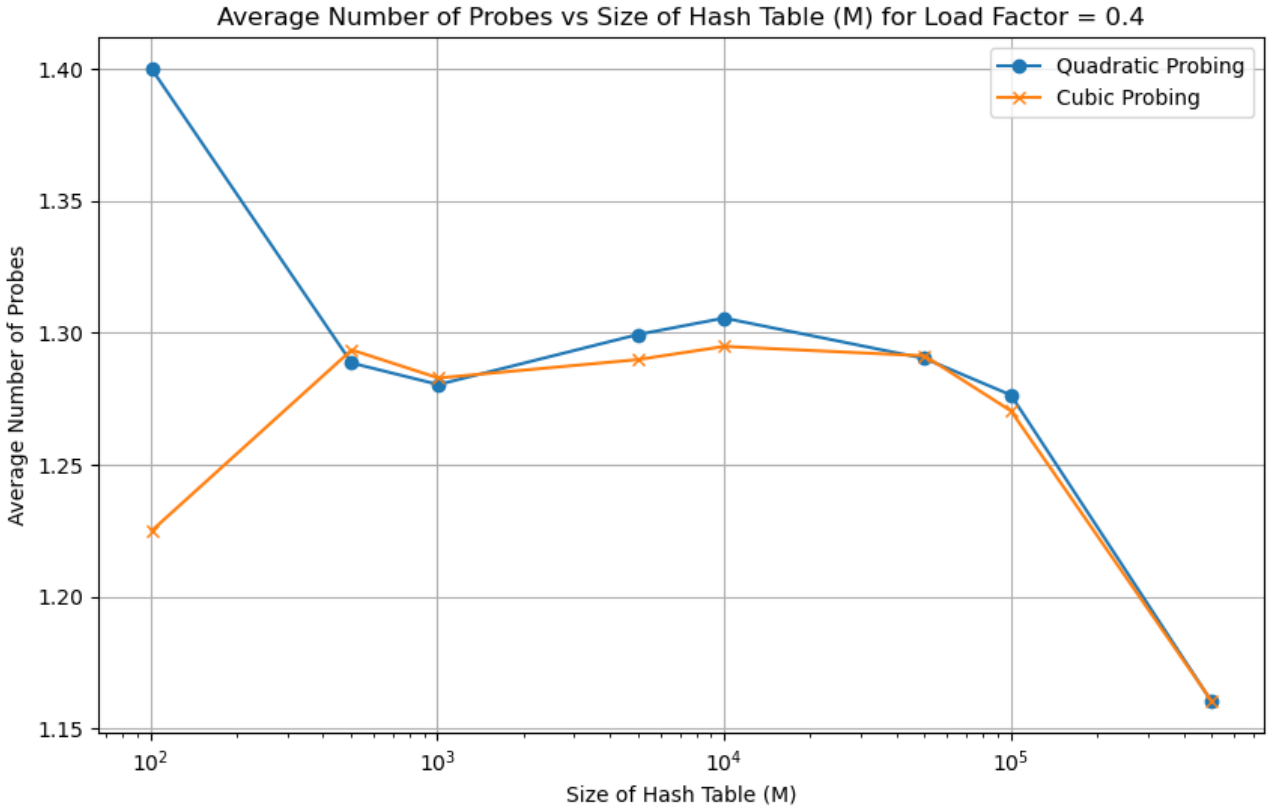

Рис. 3: Зависимость среднего числа проб при вставке от размера хеш-таблицы при малом коэффициенте заполненности

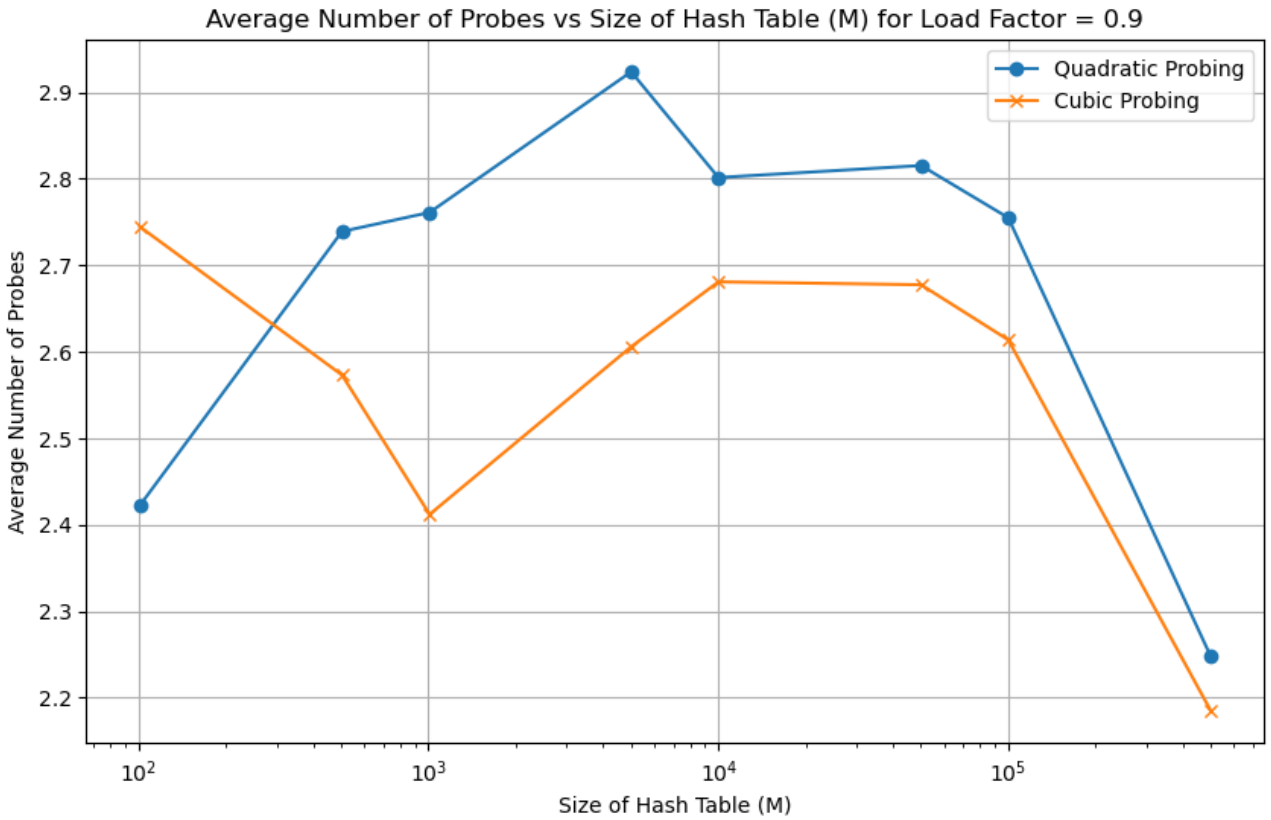

Рис. 4: Зависимость среднего числа проб при вставке от размера хеш-таблицы при большом коэффициенте заполненности

Можно заметить, что при малом коэффициенте заполненности разница между квадратичным и кубическим пробированием незначительна, но при большом коэффициенте заполненности кубическое пробирование показывает себя лучше. Это подтверждает наши предположения из предыдущего пункта.

Также при проведении эксперимента при достаточно маленьких $50 < M < 100$ происходило зацикливание при вставке элементов в хеш-таблицу методом кубического пробирования, что говорит о его недостаточной эффективности и некорректности при малых размерах хеш-таблицы (возможно, что при правильно подобранных коэффициентах c_1, c_2, c_3 такого не произошло, но тут же упор сделан на $M > 100$).