

Задача А2. Кубическое пробирование

Хеш-таблицы с открытой адресацией используют различные методы пробирования для разрешения коллизий, к основным из которых можно отнести:

1. Линейное пробирование, при котором последовательно проверяются ячейки хеш-таблицы с индексами $\text{hash}(\text{key})$, $\text{hash}(\text{key}) + 1$, $\text{hash}(\text{key}) + 2$, ...
2. Квадратичное пробирование $\text{hash}(\text{key}, i) = \text{hash}(\text{key}) + c_1 \cdot i + c_2 \cdot i^2$, при котором:
 - в простом варианте при $c_1 = c_2 = 1$ последовательно проверяются ячейки $\text{hash}(\text{key})$, $\text{hash}(\text{key})+1$, $\text{hash}(\text{key})+2$, $\text{hash}(\text{key})+6$, ...
 - для хеш-таблицы размера $M = 2^m$ при $c_1 = c_2 = 0.5$ последовательно проверяются ячейки $\text{hash}(\text{key})$, $\text{hash}(\text{key})+1$, $\text{hash}(\text{key})+3$, $\text{hash}(\text{key})+6$, ...

Мы решили пойти дальше и рассмотреть кубическое пробирование, при котором проверка ячеек в хештаблице выполняются по следующему правилу: $\text{hash}(\text{key}, i) = \text{hash}(\text{key}) + c_1 \cdot i + c_2 \cdot i^2 + c_3 \cdot i^3$.

Оцените, будет ли кубическое пробирование выполнять распределение ключей по хеш-таблице лучше (более равномерно), чем квадратичное, с точки зрения образования кластеров и возникновения коллизий. Подкрепите свои рассуждения программными экспериментами с хеш-таблицами различных размеров, а также приложите код. Ограничений на используемые языки программирования в этом задании нет.

1. Общий анализ, сравнение и обоснование достоинств, а также недостатков кубического пробирования.

Квадратичное пробирование:

Плюсы:

- Избежание первичных кластеров. В отличие от линейного пробирования, квадратичный сдвиг не приводит к быстрому росту кластеров
- Вычисление i и i^2 требует меньших вычислительных затрат чем i^3
- При правильном выборе коэффициентов и размера таблицы, например, таблица простого размера или размером 2^m с соответствующей

нормировкой коэффициентов, можно доказать, что алгоритм гарантированно исследует все ячейки или достаточно большое их количество

Минусы:

- Вторичные кластеры. Если два ключа имеют одинаковое начальное значение $\text{hash}(\text{key})$, то их последовательности проб совпадают, при фиксированных c_1, c_2 . Это приводит к вторичному кластерированию, когда разные ключи, стартуя из одной ячейки, всегда двигаются по одному и тому же набору позиций
- Ограниченное покрытие. При некоторых сочетаниях коэффициентов и размеров таблицы не гарантируется, что последовательность проб охватывает всю таблицу

Кубическое пробирование:

Плюсы:

- Добавление кубического слагаемого обеспечивает более разнообразные смещения при поиске свободных ячеек. Это означает, что даже если два ключа дают одинаковое значение $\text{hash}(\text{key})$, их последовательности проб могут оказаться различными, особенно когда используются разные параметры или осуществляется смешивание начального значения с дополнительными величинами. В результате, вероятность образования вторичных кластеров снижается
- Благодаря нерегулярности последовательности проб, кубическое пробирование может способствовать более равномерному распределению вставляемых ключей по таблице, то есть уменьшить вероятность образования кластеров

Минусы:

- Усложнение вычислений. Добавление кубического члена требует дополнительных арифметических операций, что может быть критично для систем с ограничениями по времени

- Проблемы с покрытием таблицы. Как и в квадратичном пробировании, при неверном выборе коэффициентов c_1, c_2, c_3 и размера таблицы можно столкнуться с ситуацией, когда не все ячейки будут достигнуты. Выбор параметров становится более тонким

Вывод:

Кубическое пробирование может обеспечить более равномерное распределение ключей в некоторых случаях, особенно при высокой заполненности таблицы, но за счёт возросшей вычислительной сложности и трудностей в теоретическом анализе его преимущества не являются категорически существенными по сравнению с квадратичным пробированием.

2. Реализация и анализ программных экспериментов по применению кубического пробирования для разрешения коллизий.

Код программы:

https://github.com/artishokq/AlgorithmsAndDataStructures/blob/main/SET-5/A2_experiment.cpp

Анализ результатов эксперимента:

Среднее число коллизий при вставке ключей

Эксперимент для размера таблицы 100 и количества ключей 80:

Квадратичное пробирование: среднее число коллизий = 92.68

Кубическое пробирование: среднее число коллизий = 83.62

Эксперимент для размера таблицы 200 и количества ключей 160:

Квадратичное пробирование: среднее число коллизий = 182.50

Кубическое пробирование: среднее число коллизий = 175.00

Эксперимент для размера таблицы 500 и количества ключей 400:

Квадратичное пробирование: среднее число коллизий = 466.38

Кубическое пробирование: среднее число коллизий = 452.46

Можно сделать вывод, что кубическое пробирование может выполнять распределение ключей несколько более равномерно, особенно в ситуациях высокой заполненности таблицы, но в большинстве практических сценариев его преимущества не перевешивают дополнительные вычислительные затраты и сложность реализации по сравнению с квадратичным пробированием.