

Тепляков Владислав БПИ2310

SET 5

Задача A1

Часть 1

Проблема приведенных реализаций заключается в следующем: алгоритм вставки игнорирует ячейки с меткой ERASED, вставляя новые ключи только в первую найденную NULL-ячейку. Это приводит к увеличению размера кластеров, что ведет к длительным проверкам при поиске и вставке. Пример последовательности операций:

- Допустим, ключи A, B, C при хешировании по модулю $M = 5$ попадают в ячейку с индексом 0. Тогда после вставки этих значений хеш-таблица будет выглядеть следующим образом: **[A, B, C, NULL, NULL]**
- Теперь произведем удаление элемента B: **[A, ERASED, C, NULL, NULL]**
- Далее вставим новый ключ D, который также захешировался в ячейку с индексом 0. A; ERASED, но сюда не вставляем; C; NULL, вставляем в ячейку с индексом 3.

Итоговая таблица выглядит так: **[A, ERASED, C, D, NULL]**

Часть 2

Чтобы избежать такой проблемы, изменим алгоритм INSERT так, чтобы он заполнял первую доступную ячейку NULL или ERASED:

=====

INSERT(key):

ind = hash(key) mod M

while (table[ind] != NULL)

if (table[ind] == key)

return

if (table[ind] == ERASED)

break

ind = (ind + 1) mod M

table[ind] = key

=====

Теперь при вставке, если на пути была встречена ячейка ERASED, в нее и будет записано значение.