

Хеш-таблиця

Лаворик Ольга

Що таке асоціативний масив

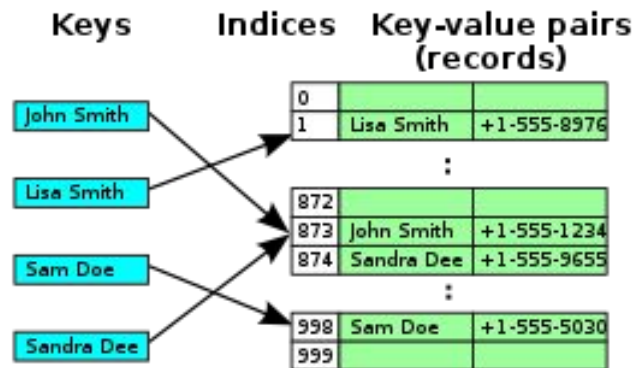
Асоціативний масив – тип даних, що дозволяє зберігати інформацію у вигляді пари ключ-значення. Також підтримує операції (виконуються за час $O(1)$) :

- ❑ Додати елемент – INSERT(ключ, значення)
- ❑ Знайти елемент– FIND(ключ)
- ❑ Видалити елемент– REMOVE(ключ)

Одна із реалізацій – хеш-таблиця. Наприклад, У мові C++ в STL реалізовано контейнер map.

Хеш-функції

- Вираховується хеш-функція від ключа
- $i = \text{hash}(\text{key})$ - індекс масиву
- Операція перенаправляється до об'єкта `array[i]`



Колізії

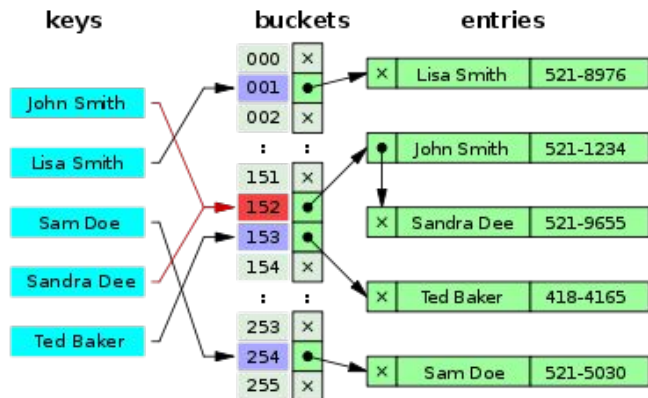
— — —

Ідеальна хеш функція –різні ключі отримують різні значення хешу(індексу). На практиці це не завжди можливо, тому виникають колізії. Хеш-таблиці мають методи вирішення колізій

Метод ланцюжків

— — —

Комірка масиву – вказівник
на зв'язний список пар ключ-
значення.

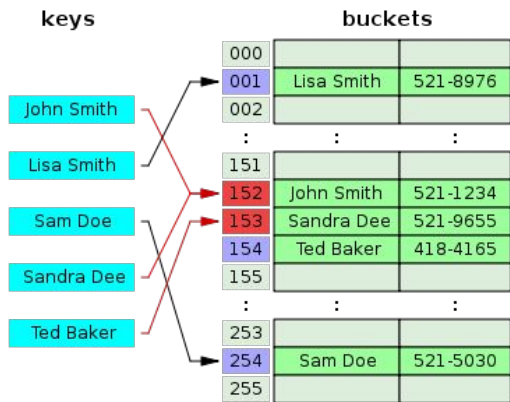


```
class LinkHashEntry {
private:
    int key;
    int value;
    LinkHashEntry *next;
public:
    LinkHashEntry(int key, int value) {
        this->key = key;
        this->value = value;
        this->next = NULL;
    }
    int getKey() {
        return key;
    }
    int getValue() {
        return value;
    }
    void setValue(int value) {
        this->value = value;
    }
    LinkHashEntry *getNext() {
        return next;
    }
    void setNext(LinkHashEntry *next) {
        this->next = next;
    }
};
```

Відкрита адресація

— — —

В масиві зберігаються пари ключ-значення. Алгоритм вставки перевіряє комірки масива, поки не знайде першу вільну.



```
class HashEntry {
private:
    int key;
    int value;
public:
    HashEntry(int key, int value) {
        this->key = key;
        this->value = value;
    }

    int getKey() {
        return key;
    }
    int getValue() {
        return value;
    }
    void setValue(int value) {
        this->value = value;
    }
};
```