Хеш-таблица. Открытая адресация

Гусев Илья, Булгаков Илья

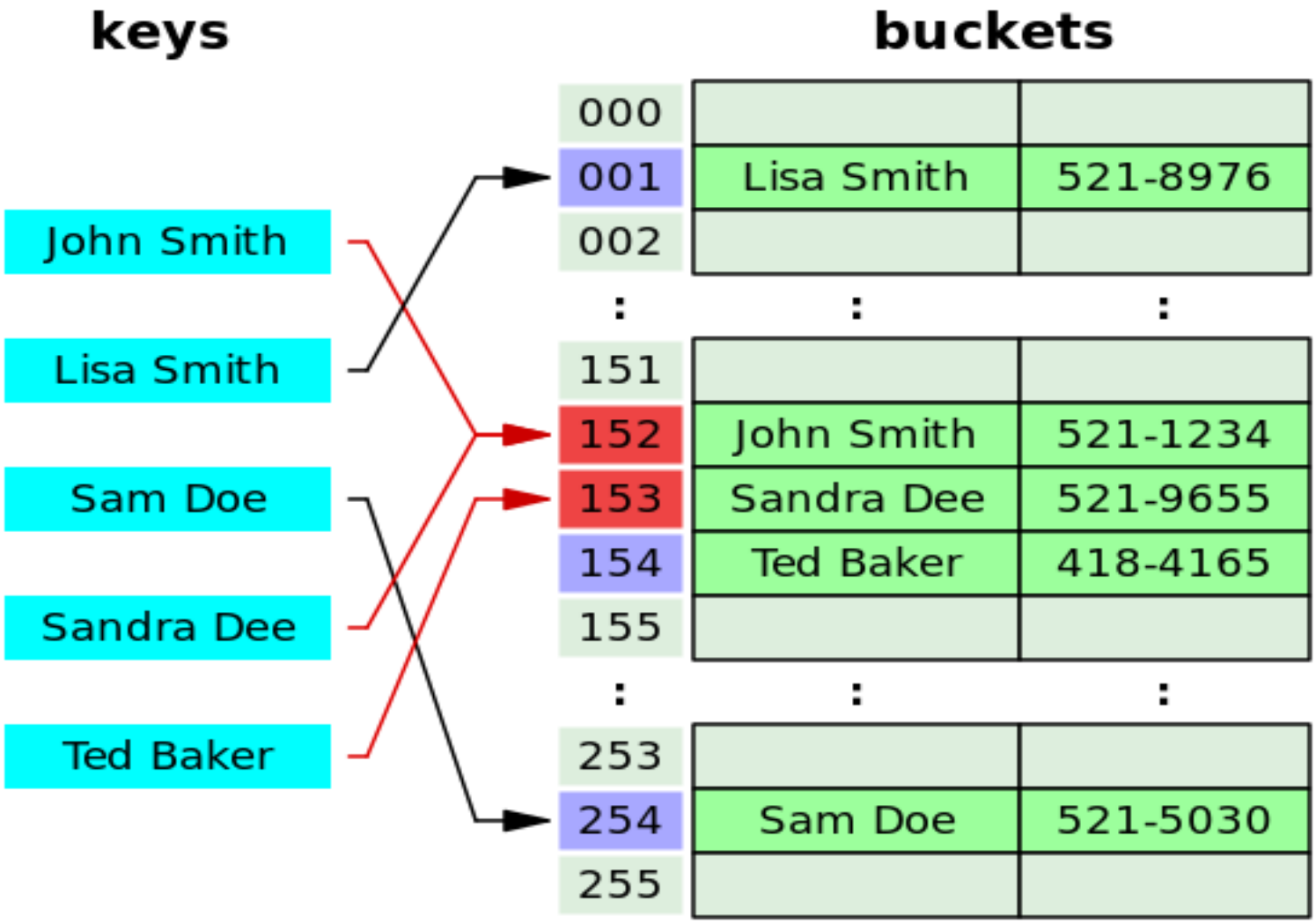
Московский физико-технический институт

Москва, 2018

Содержание

- Открытая адресация
- Методы пробирования

Открытая адресация



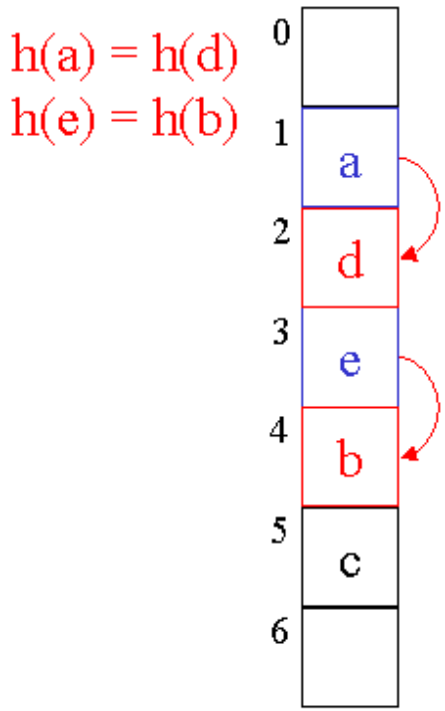
Открытая адресация

Closed Hashing *or* Open Addressing

What if we only allow one Key at each entry?

- two objects that hash to the same spot can't both go there
- first one there gets the spot
- next one must *go in another spot*

- Properties
  - $\lambda \leq 1$
  - performance degrades with difficulty of finding right spot



Методы пробирования

В случае появления коллизий надо придумать, в каком порядке просматривать ячейки. Есть несколько подходов.

- Линейное пробирование
  $H(k, i) = (Hash(k) + i) mod(m)$
- Квадратичное пробирование
  $H(k, i) = (Hash(k) + C1 * i + C2 * i^2) mod(m)$
  Например, C1 = C2 = 1/2
- Двойное хеширование
  $H(k, i) = (Hash1(k) + i * Hash2(k)) mod(m)$

Линейное пробирование

Linear Probing

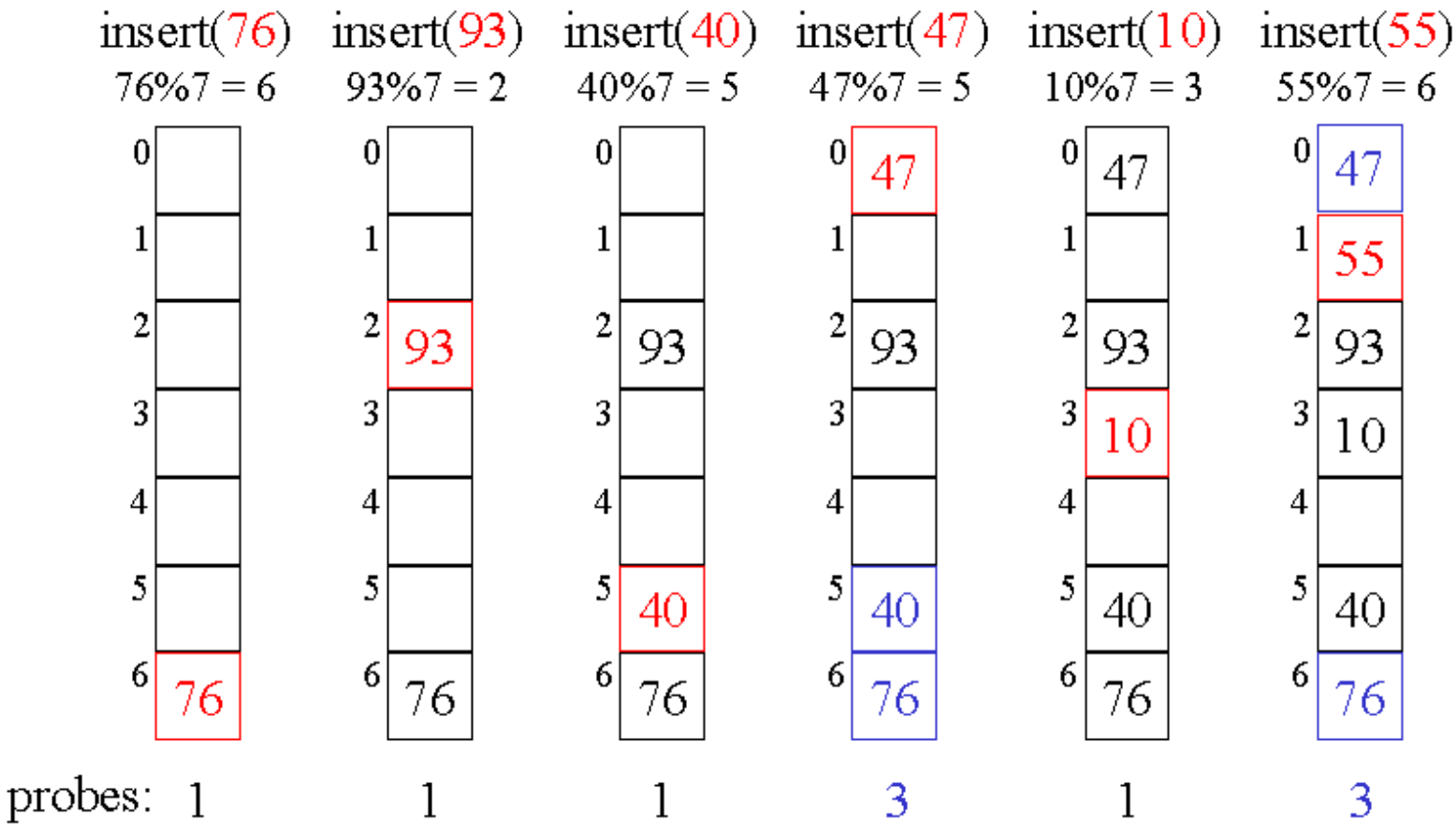
f(i) = i

- Probe sequence is
  - h(k) mod size
  - h(k) + 1 mod size
  - h(k) + 2 mod size
  - …
- findEntry using linear probing:

```
bool findEntry(const Key & k, Entry *& entry) {
  int probePoint = hash1(k);
  do {
    entry = &table[probePoint];
    probePoint = (probePoint + 1) % size;
  } while (!entry->isEmpty() && entry->key != k);
  return !entry->isEmpty();
}
```

Линейное пробирование

Linear Probing Example



Квадратичное пробирование

Quadratic Probing

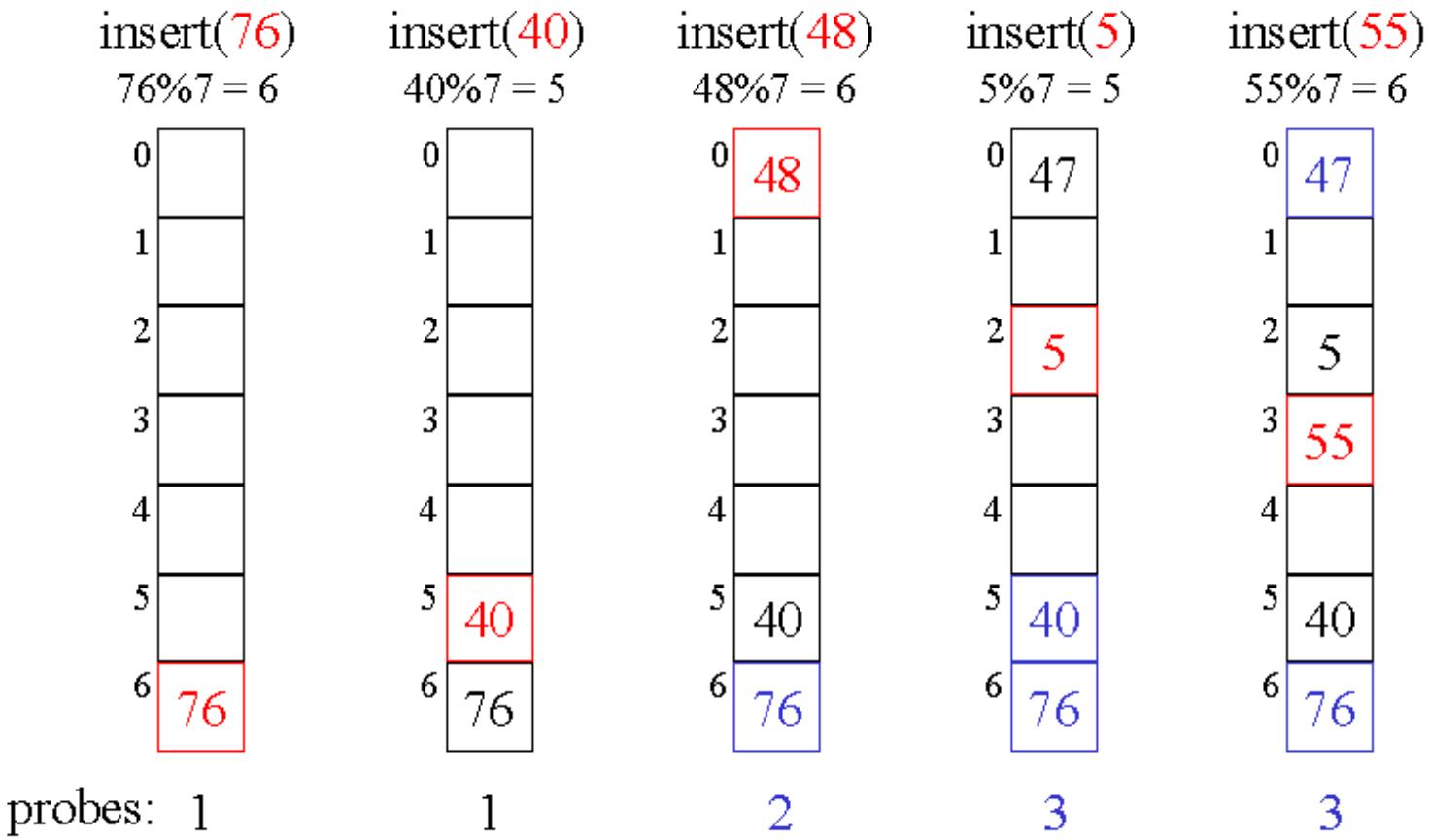
$f(i) = i^2$

- Probe sequence is
  - h(k) mod size
  - (h(k) + 1) mod size
  - (h(k) + 4) mod size
  - (h(k) + 9) mod size
  - …
- findEntry using quadratic probing:

```
bool findEntry(const Key & k, Entry *& entry) {
  int probePoint = hash1(k), numProbes = 0;
  do {
    entry = &table[probePoint];
    numProbes++;
    probePoint = (probePoint + 2*numProbes - 1) % size;
  } while (!entry->isEmpty() && entry->key != key);
  return !entry->isEmpty();
}
```

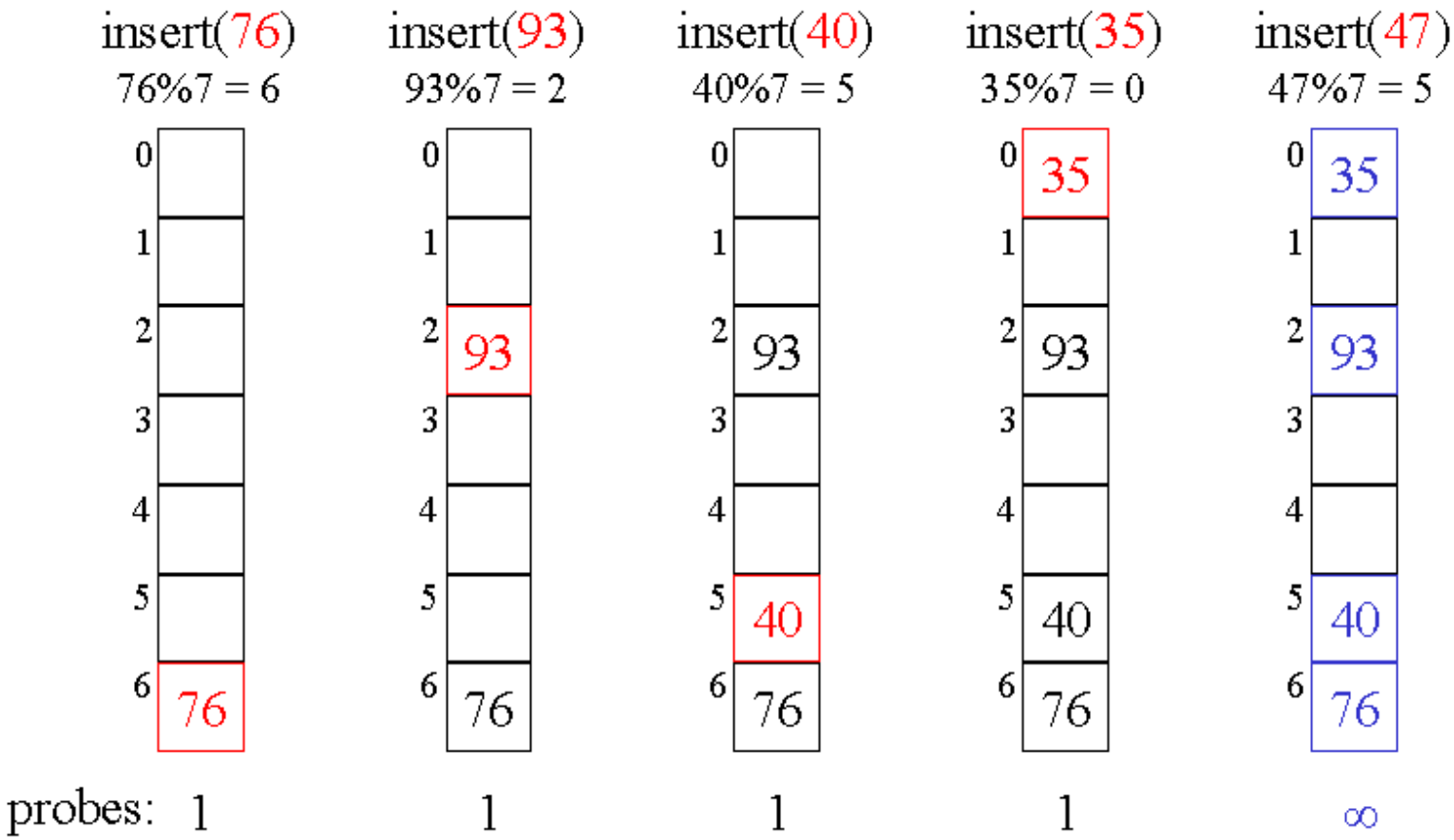
Квадратичное пробирование

Quadratic Probing Example ☺



Квадратичное пробирование

Quadratic Probing Example ☹



Двойное хеширование

Double Hashing

$f(i) = i \cdot hash_2(x)$

- Probe sequence is
  - $h_1(k) \bmod size$
  - $(h_1(k) + 1 \cdot h_2(x)) \bmod size$
  - $(h_1(k) + 2 \cdot h_2(x)) \bmod size$
  - ...
- Code for finding the next linear probe:

```
bool findEntry(const Key & k, Entry *& entry) {
  int probePoint = hash1(k), hashIncr = hash2(k);
  do {
    entry = &table[probePoint];
    probePoint = (probePoint + hashIncr) % size;
  } while (!entry->isEmpty() && entry->key != k);
  return !entry->isEmpty();
}
```

Двойное хеширование