Rehashing

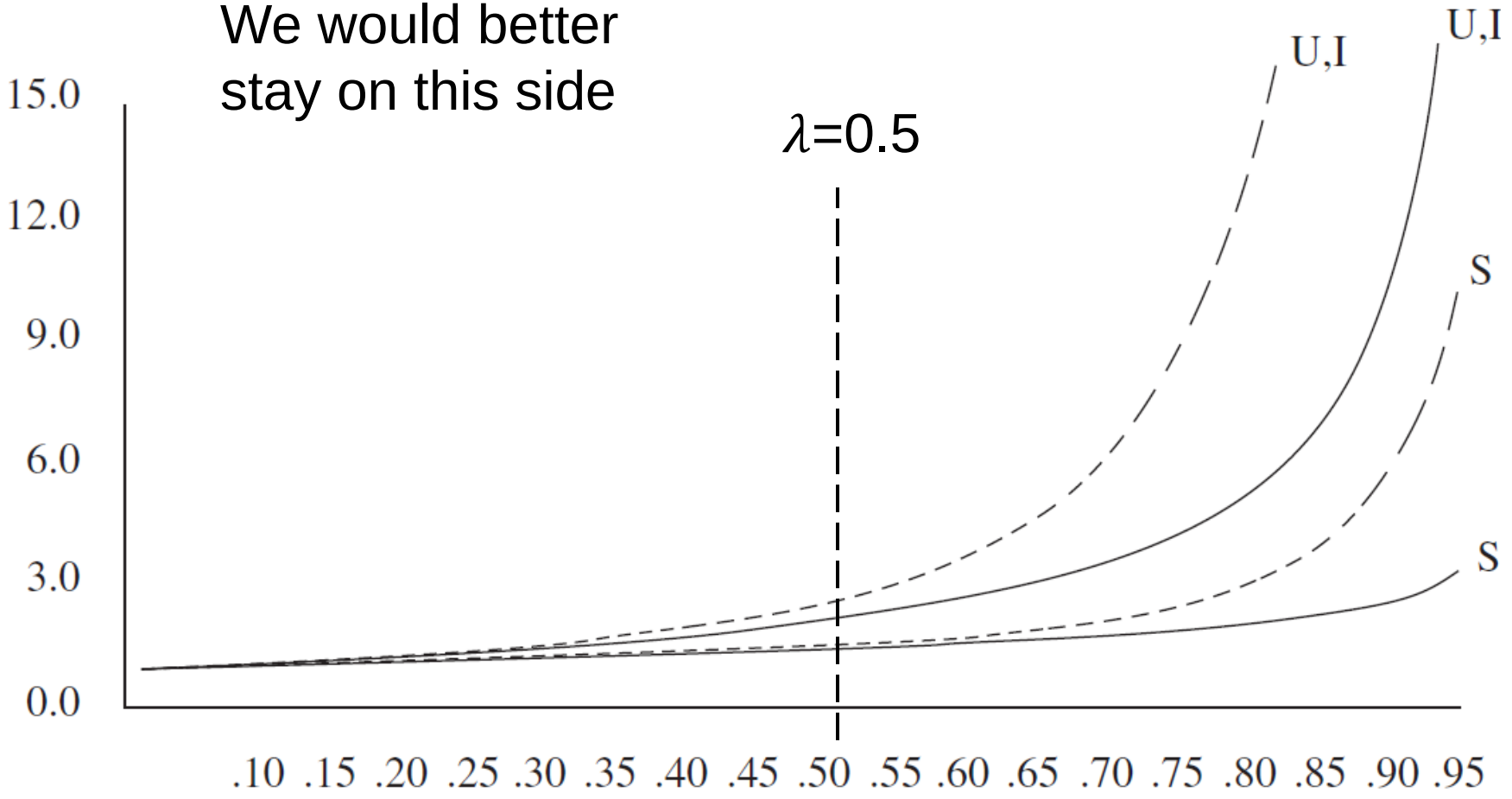
- As more keys are inserted into the hashtable, the performance degrades. Why?
- The solution to this problem is *rehashing*. A new hashtable is created, and all keys are rehashed to the new table.
- Q1: When is a good time to rehash?
- Q2: What is a good size for the new hashtable?

When to rehash?

- In arrays, we used to expand the array when it is full. Should we do the same with hashtables?
- What happens if we wait until the hashtable is full?
  - with linear probing
  - with quadratic probing
- Load Factor: $\lambda = \frac{\#\ of\ keys}{\#\ of\ buckets}$
- $0 \leq \lambda \leq 1$

Load Factor

We would better stay on this side

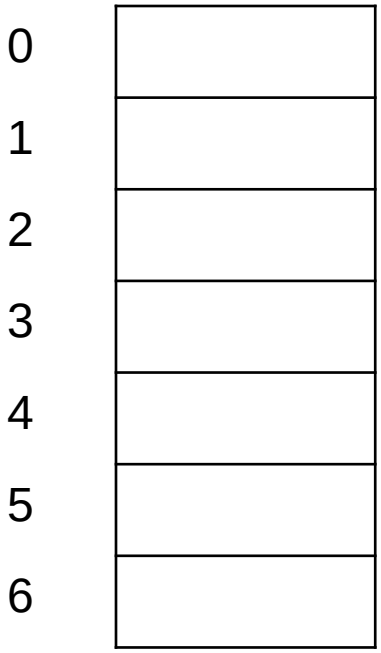


Rehashing

- When $\lambda > 0.5$
- New size is roughly double the old side

```
Rehash() {
  T* newHashtable = new T[new_size];
  for (i = 0 to old_size) {
    if (bucket #i is occupied) {
      Insert the key at bucket #i into the new table;
    }
  }
  replace the old table with the new one;
}
```

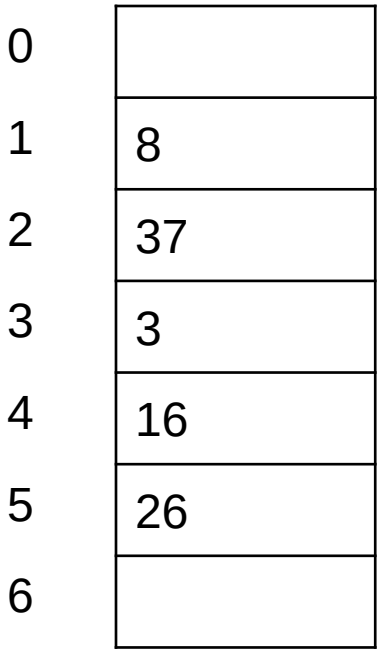
Rehashing Example



h = x % 7

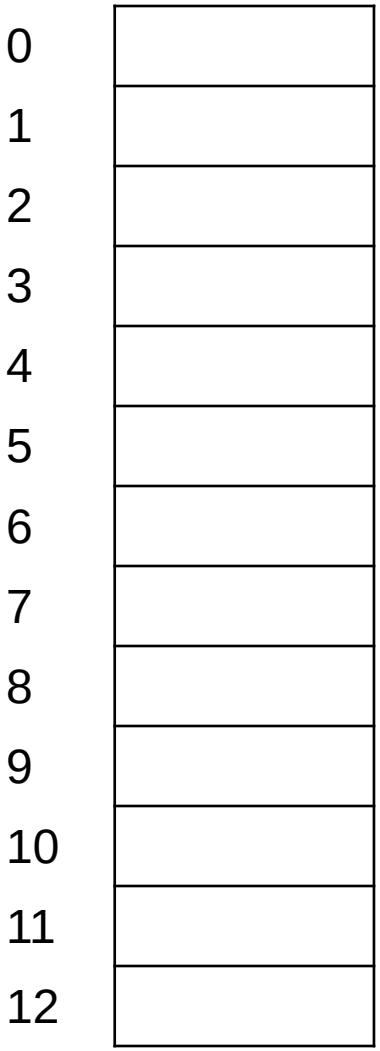
Insert {37, 8, 3, 16, 26}

Rehashing Example



h = x % 7

Insert {30, 8, 3, 16, 26}



h = x % 13

Rehashing Example

0	
1	8
2	37
3	3
4	16
5	26
6	

h = x % 7

Insert {37, 8, 3, 16, 26}

0	26
1	
2	
3	3
4	16
5	
6	
7	
8	8
9	
10	
11	37
12	

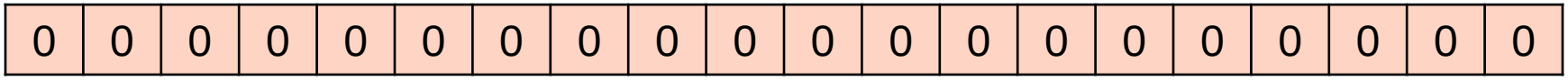
h = x % 13

Application to Hashtables

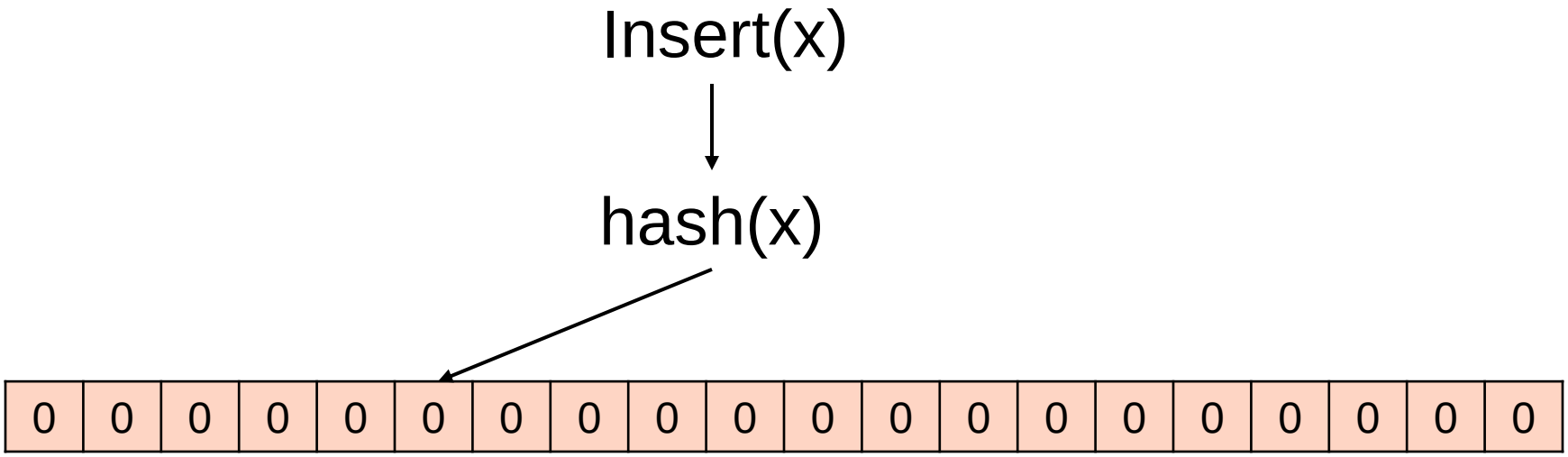
- Bloom filter
- Stores a set of keys
- Answers one question: Is the key x in the set or not?
- Application: Used as a prefilter to avoid costly searches when the key is not there
  - e.g., BST search, hashtable search, ordered list search, unordered list search

Bloom Filter

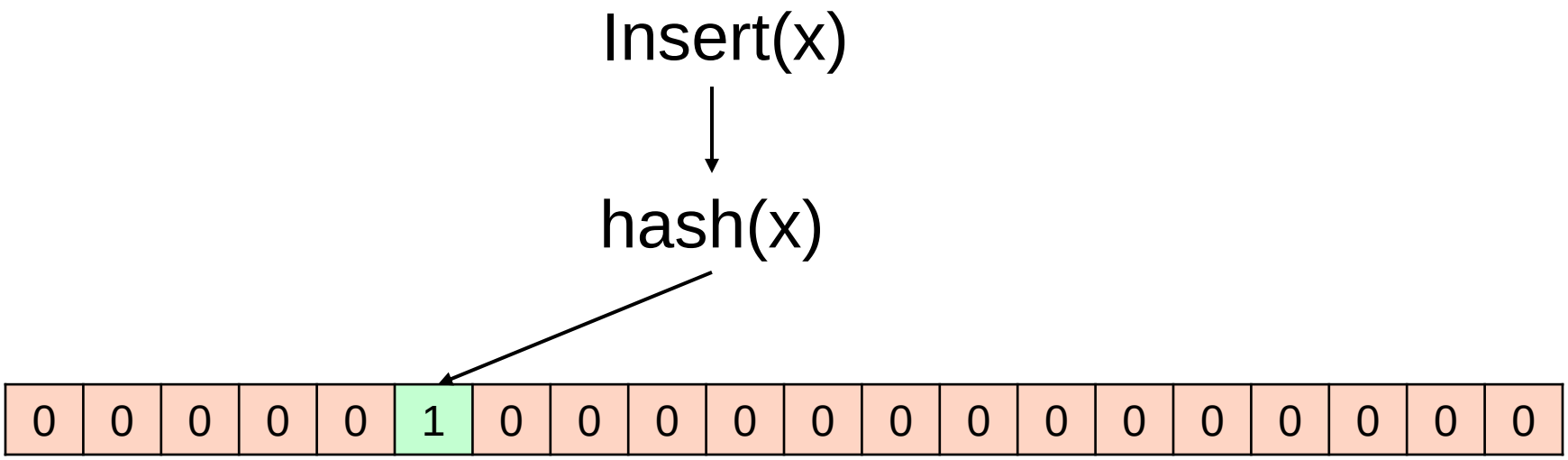
Initialize: Create a bit vector all set to zeros



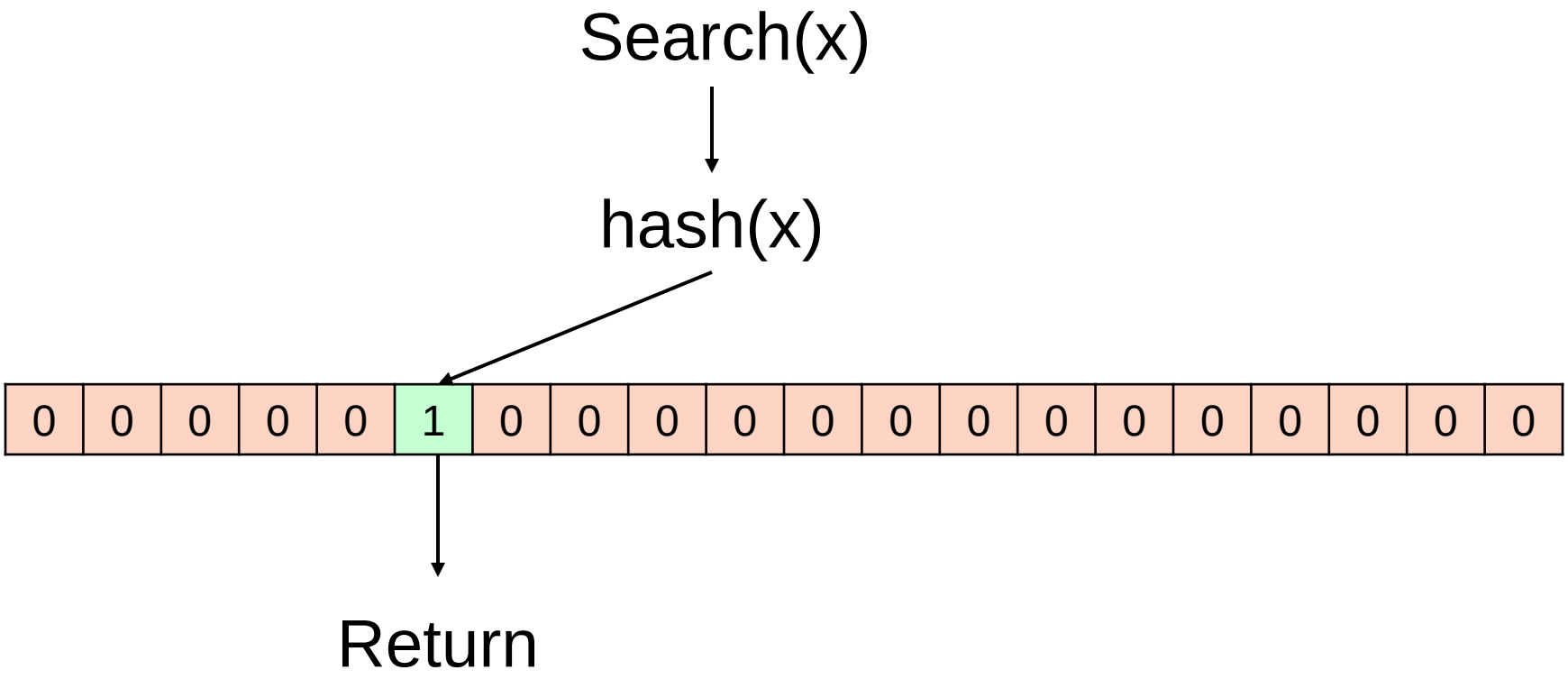
Bloom Filter



Bloom Filter



Bloom Filter



Implementation

```
Initialize(m) {
  b = new bit vector[m];
}
```

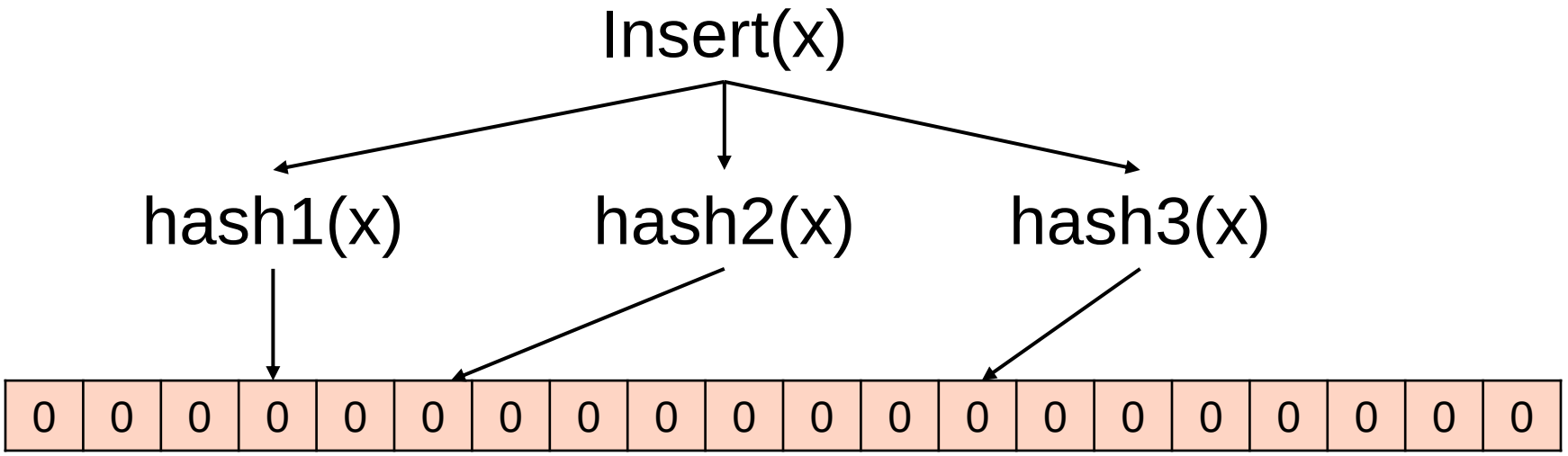
```
Insert(x) {
  b[hash(x)] = 1;
}
```

```
Search(x) {
  return b[hash(x)];
}
```

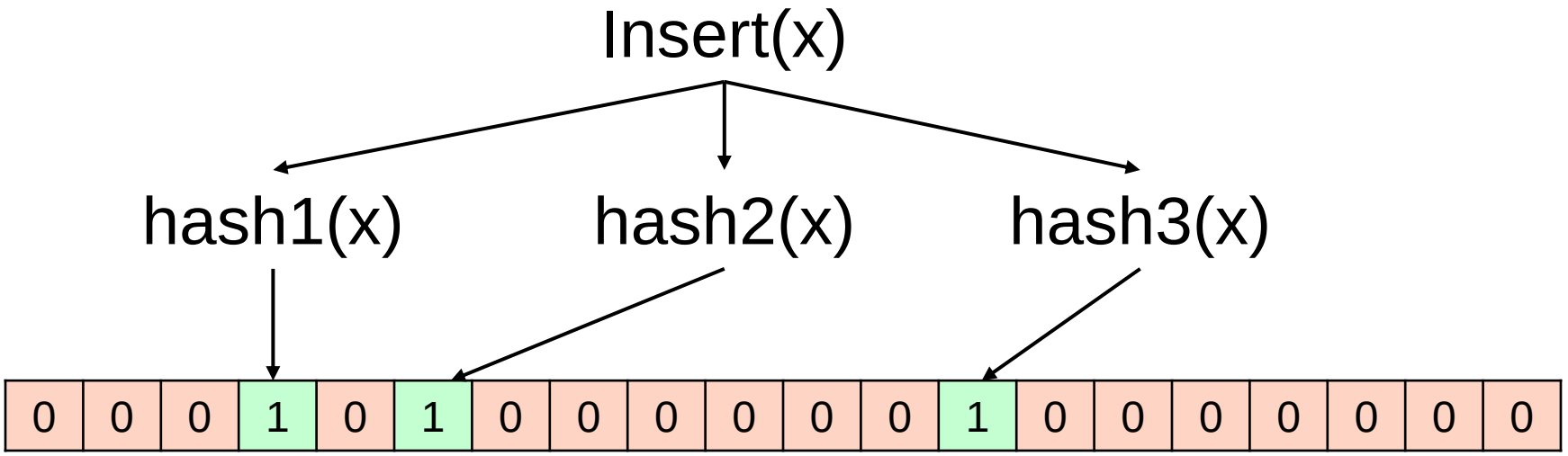
Collisions

- What to do with collisions?
- Nothing!!
- What are the consequences of this?
- False positives
- How to support deletions?
- Deletions are not supported

Multiple Hash Functions



Multiple Hash Functions



Multiple Hash Functions