**Goal of hash tables is to achieve O(1) for searching**

**Basic idea about hash table**

**To provide a key for your interested group of values and save the keys and values at location in the array Table[key].**

Ting 1

Jiang 2

Jane 5

Timothy 6

Bob 8

Green 9

Table [10]

|  |
| --- |
|  |
| 1 Ting |
| 2 Jiang |
|  |
|  |
| 5 Jane |
| 6 Timothy |
|  |
| 8 Bob |
| 9 Green |

After inserting values, if we start to do searching, it is very fast!

Search (Bob)

8

Table[8]

Search(Hello) – 7

7

Table[7]

**Basic idea about hash functions**

**To map the keys to hashes with modulus to reduce potential memory waste. Hash functions are in the form of int hash=key%table\_size.**

Ting 1

Jiang 2

Jane 5

Timothy 6

Bob 8

Green 10000000

**Hash Functions**

10000000%10= 0

Table[a]

Table[0]

|  |
| --- |
| 10000000 Green |
| 1 Ting |
| 2 Jiang |
|  |
|  |
| 5 Jane |
| 6 Timothy |
|  |
| 8 Bob |
|  |

If a=1

Table[1];

Example:

Ting 1

Jiang 2

Jane 5

Timothy 6

Bob 8

Green 10000001

**Hash Functions**

10000001%10= 1

Table[a]

Table[1]

|  |
| --- |
|  |
| 1 Ting |
| 2 Jiang |
|  |
|  |
| 5 Jane |
| 6 Timothy |
|  |
| 8 Bob |
|  |

**Collision**

**Green will overwrite Ting !!!**

* **Linear Probing**

|  |
| --- |
|  |
| 1 Ting |
| 2 Jiang |
| 10000001 Green |
|  |
| 5 Jane |
| 6 Timothy |
|  |
| 8 Bob |
|  |

**Clustering**

|  |
| --- |
|  |
| 1 Ting |
| 2 Jiang |
| 1000001 Green |
| 1001 Steve |
| 5 Jane |
| 6 Timothy |
|  |
| 8 Bob |
|  |

* **Quadratic Probing**
  + **Probe sequence from location *i* is  
    *i + 1, i – 1, i + 4, i – 4, i + 9, i – 9*, …**
* **Double hashing** 
  + **Use a second hash function to determine probe sequence**
* **Chaining** 
  + **Table is a list or vector of head nodes to linked lists**
  + **When item hashes to location, it is added to that linked list**

**About Strings – How to compute the keys**

Ting 1

Jiang 2

Jane 5

Timothy 6

Bob 8

Green 10000000

* **Possible to manipulate numeric (ASCII) value of first and last characters of a name**