Hash Functions

Kuan-Yu Chen (陳冠宇)

2020/12/23 @ TR-212, NTUST

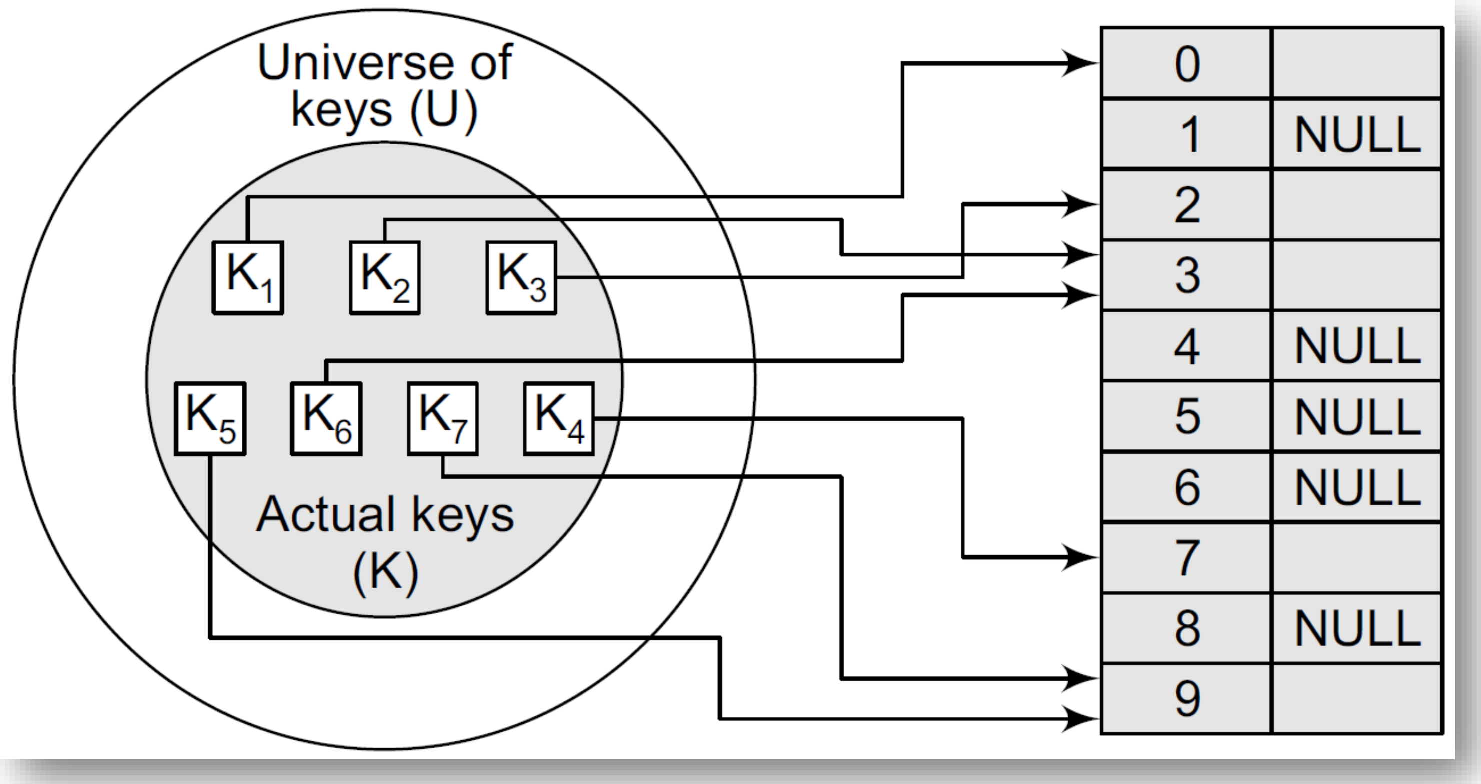
Review

- Graph is an important data structure

	Undirected Graph	Directed Graph
Definition	$G = (V, E)$	
Node Degree	$\deg(u)$	$\deg(u) = indeg(u) + outdeg(u)$
Representation	Sequential Representation Linked Representation Adjacency Multi-list	
Search	Breadth-first Search (Queue) Depth-first Search (Stack)	
Traverse	Breadth-first Search (Queue) Depth-first Search (Stack)	
Minimal Spanning Tree	Prim's Algorithm Kruskal's Algorithm	
Shortest Path		Dijkstra's Algorithm Bellman-ford Algorithm

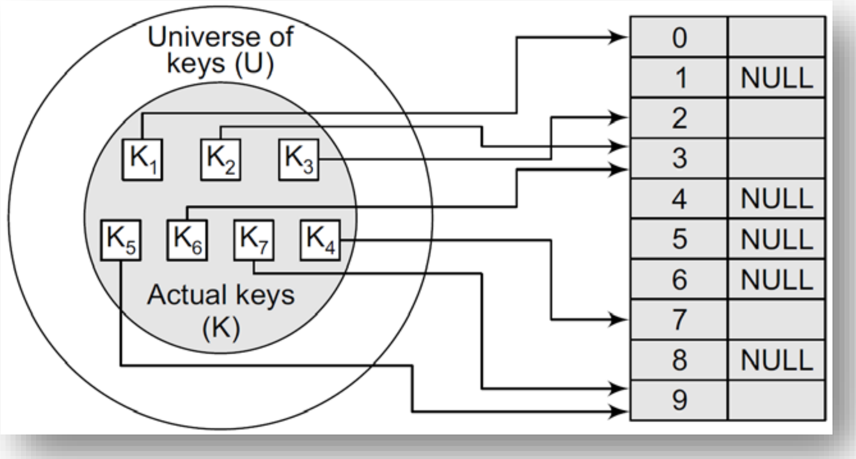
Hashing.

- Hash table is a data structure in which keys are mapped to array positions by a hash function
  - An element with key k is stored at index $h(k)$
    - It means a hash function h is used to calculate the index at which the element with key k will be stored

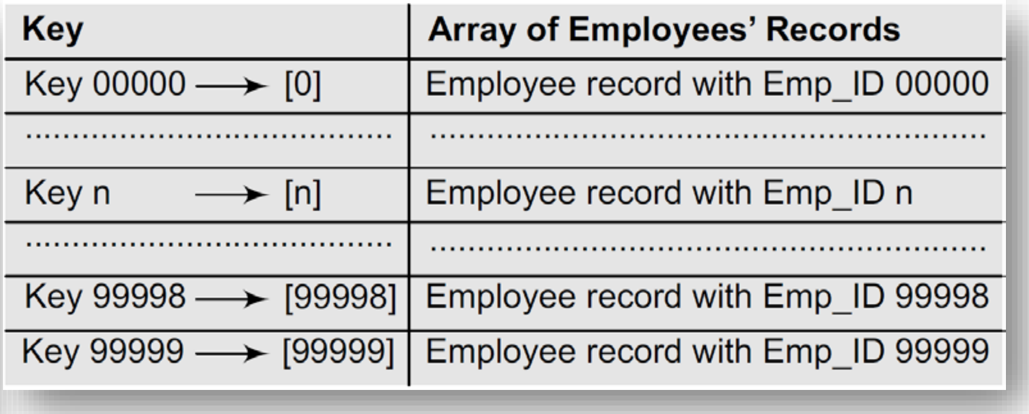


Hashing..

- The main goal of using a hash function is to reduce the range of array indices that have to be handled
  - Instead of having U values, we just need K values, thereby reducing the amount of storage space required
    - Given key values range from 00000 to 99999
    - We need an array of size 100,000, but only 100 elements will be used



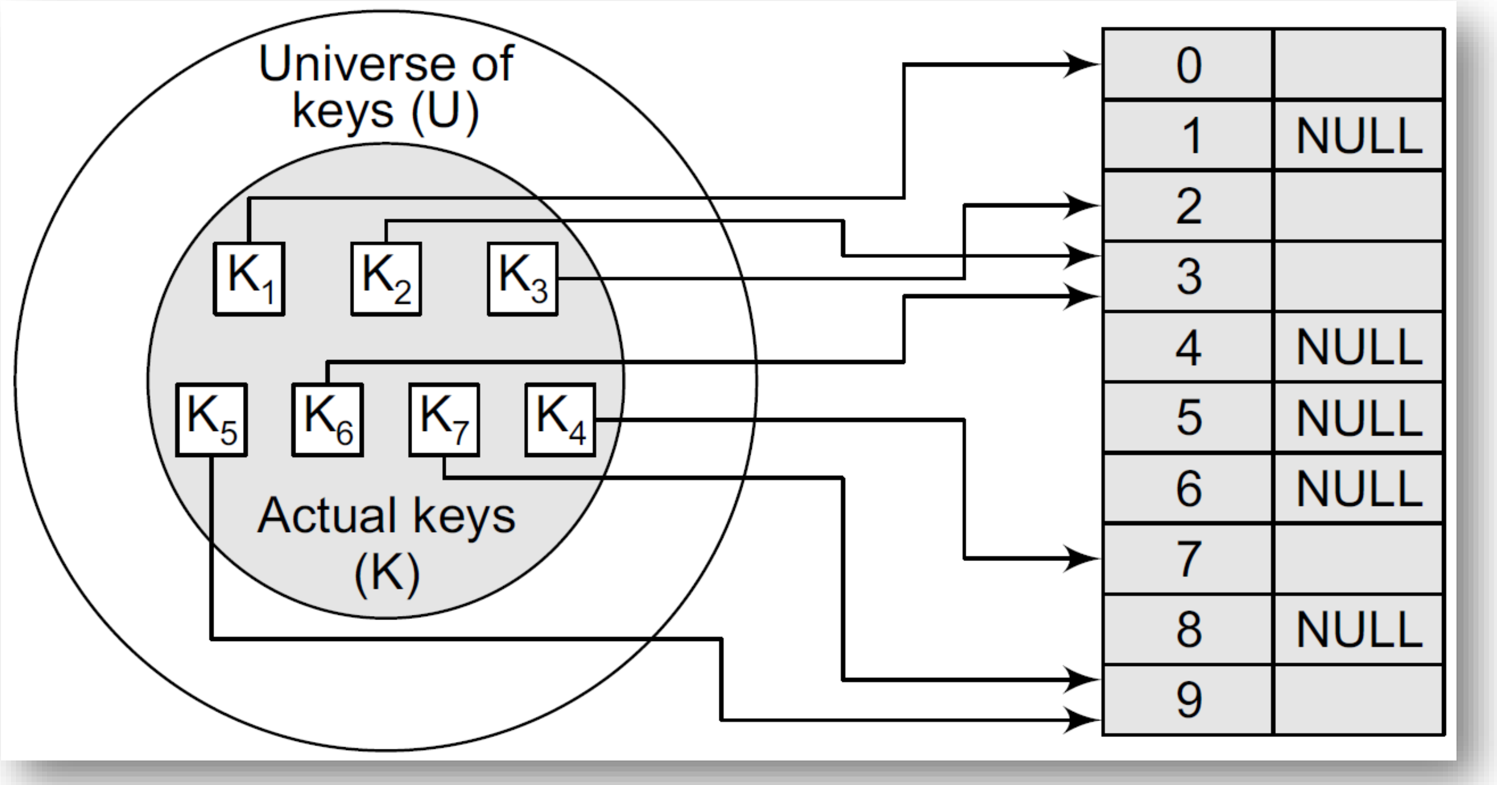
0	
1	NULL
2	
3	
4	NULL
5	NULL
6	NULL
7	
8	NULL
9	



Key	Array of Employees' Records
Key 00000 → [0]	Employee record with Emp_ID 00000
....	....
Key n → [n]	Employee record with Emp_ID n
....	....
Key 99998 → [99998]	Employee record with Emp_ID 99998
Key 99999 → [99999]	Employee record with Emp_ID 99999

Hashing...

- When two or more keys map to the same memory location, a **collision** is said to occur
  - k_2 and k_6
  - k_5 and k_7



Hash Functions.

- A hash function is a **mathematical formula** which, when applied to a key, produces an integer which can be used as an index for the key in the hash table
  - It produces a unique set of integers within some suitable range in order to reduce the number of collisions
    - In practice, there is **no hash function that eliminates collisions completely**

Hash Functions..

- Properties of a good hash function
  - ***Low cost***
    - The cost of executing a hash function must be small
  - ***Determinism***
    - A hash procedure must be deterministic
  - ***Uniformity***
    - A good hash function must map the keys as evenly as possible over its output range
      - This means that the probability of generating every hash value in the output range should roughly be the same
      - The property of uniformity also minimizes the number of collisions

Representative Hash Functions.

- Division Method
  - It is the most simple method of hashing an integer x

$$h(x) = x \bmod M$$

    - Since it requires only a single division operation, the method works very fast
    - Extra care should be taken to select a suitable value for M
  - Suppose M is an even number then $h(x)$ is even if x is even and $h(x)$ is odd if x is odd
    - If all possible keys are equal-probable, then this is not a problem
    - If even keys are more likely than odd keys, then the division method will not spread the hashed values uniformly
    - It is usually to choose M to be a prime number

Example

- Calculate the hash values of keys 1234 and 5462 by referring to $M = 97$

$$h(1234) = 1234\%97 = 70$$

$$h(5642) = 5642\%97 = 16$$

Representative Hash Functions..

- Multiplication Method
  - The method is defined by

$$h(x) = \lfloor m(xA \bmod 1) \rfloor$$

    - Choose a constant A such that $0 < A < 1$
    - Multiply the key x by A
    - Extract the fractional part of xA
    - Multiply the fractional part of xA by the size of hash table m

  - The greatest advantage of this method is that it works practically with any value of A
    - Generally, a good choice of A is $\frac{\sqrt{5}-1}{2} = 0.618033$

Example

- Given a hash table of size 1000, map the key 12345 to an appropriate location in the hash table

$$A = 0.618033$$

$$m = 1000$$

$$\begin{aligned} h(12345) &= \lfloor m(12345 \times A \bmod 1) \rfloor \\ &= \lfloor 1000 \times (12345 \times 0.618033 \bmod 1) \rfloor \\ &= \lfloor 1000 \times (7629.617385 \bmod 1) \rfloor \\ &= \lfloor 1000 \times (0.617385) \rfloor \\ &= \lfloor 617.385 \rfloor = 617 \end{aligned}$$

Representative Hash Functions...

- Mid-Square Method
  - The algorithm works well because most or all digits of the key value contribute to the result
  - In general, the function is defined by:

$$h(x) = r - digit\left(x^2\right) = s$$

where s is obtained by selecting r digits from x^2

- Square the value of the key
- Extract the middle r digits of the result

Example

- Calculate the hash value for keys 1234 and 5642 using the mid-square method
  - The hash table has 100 memory locations
    - Note that the hash table has 100 memory locations whose indices vary from 0 to 99

      This means that only two digits are needed to map the key to a location in the hash table, so $r = 2$

$$x = 1234, \qquad x^2 = 152\underline{27}56, \qquad h(1234) = 27$$

$$x = 5642, \qquad x^2 = 3183\underline{21}64, \qquad h(5642) = 21$$

Representative Hash Functions….

- Folding Method

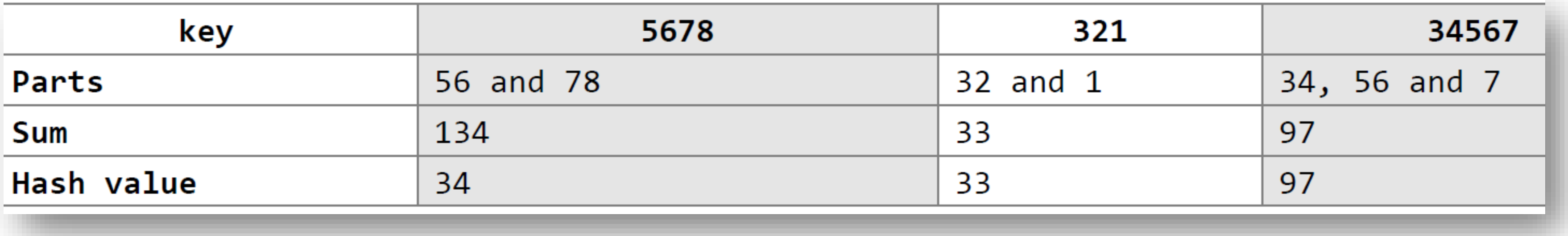
$$h(x) = r - digit\left(sum\left(divide(x)\right)\right) = s$$

  - The folding method works in the following two steps
    - Divide the key value into a number of parts
    - Add the individual parts, and extract the last r digits of the result

  - Note that the number of digits in each part of the key will vary depending upon the size of the hash table

Example

- Given a hash table of 100 locations, calculate the hash value using folding method for keys 5678, 321, and 34567
  - Since there are 100 memory locations to address, we will break the key into parts where each part (except the last) will contain two digits



key	5678	321	34567
Parts	56 and 78	32 and 1	34, 56 and 7
Sum	134	33	97
Hash value	34	33	97

Questions?

kychen@mail.ntust.edu.tw