

\* linked list: (key, value) <sup>data of node</sup> <sub>array entry</sub>

	insert	get
(unordered) key	$O(1) / O(1)$	$O(n) / O(n)$
ordered key	$O(n) / O(n)$	$O(n) / O(\log n)$
key as array index	$O(1)$	$O(1)$

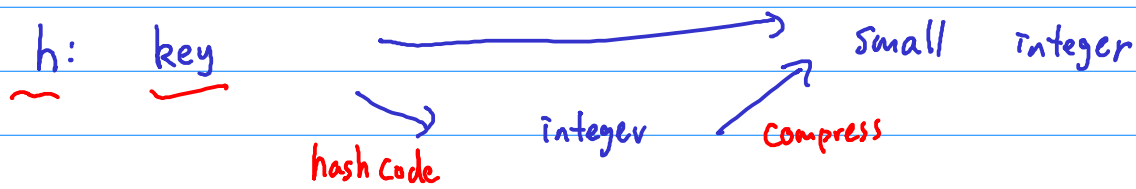
array  $\rightarrow$  integer  $\rightarrow$  key as array index, value as array data

key as array index, value as array data

\*  $\text{key} \equiv \text{integer}$   $\Rightarrow$  need super big array (space)

non-small integer  $\Rightarrow$  need super big array (space)

Compress  $\rightarrow$  array



$A.\text{insert}(\text{key}, \text{value}) \Rightarrow A[h(\text{key})] = \text{value}$   
 $v = A.\text{get}(\text{key}) \Rightarrow v = A[h(\text{key})]$

100

{key}



{0, 1, 2, ..., 99, k-1}

bucket array      hash function

"perfect" hashing

any  $h$  of same  $h(\cdot)$

\* #keys > K  
 $\Rightarrow$  two keys  
 $\Rightarrow$  Collision

$h(\text{str}) = \underline{0}$

\* key: str "apple"

$\text{HashCode}(\text{str}) = \text{str}[0] - 'a' \in \{0, 1, 2, \dots, 25\}$

$\text{Hashcode}(\text{"apple"}) = \text{Hashcode}(\text{"act"})$

$\text{Hash Code}(\text{str}) = \sum_i (\text{str}[i] - 'a')$   $2^{32}$

$\in \{0, \dots, 99\}$   
 % 100 Compress