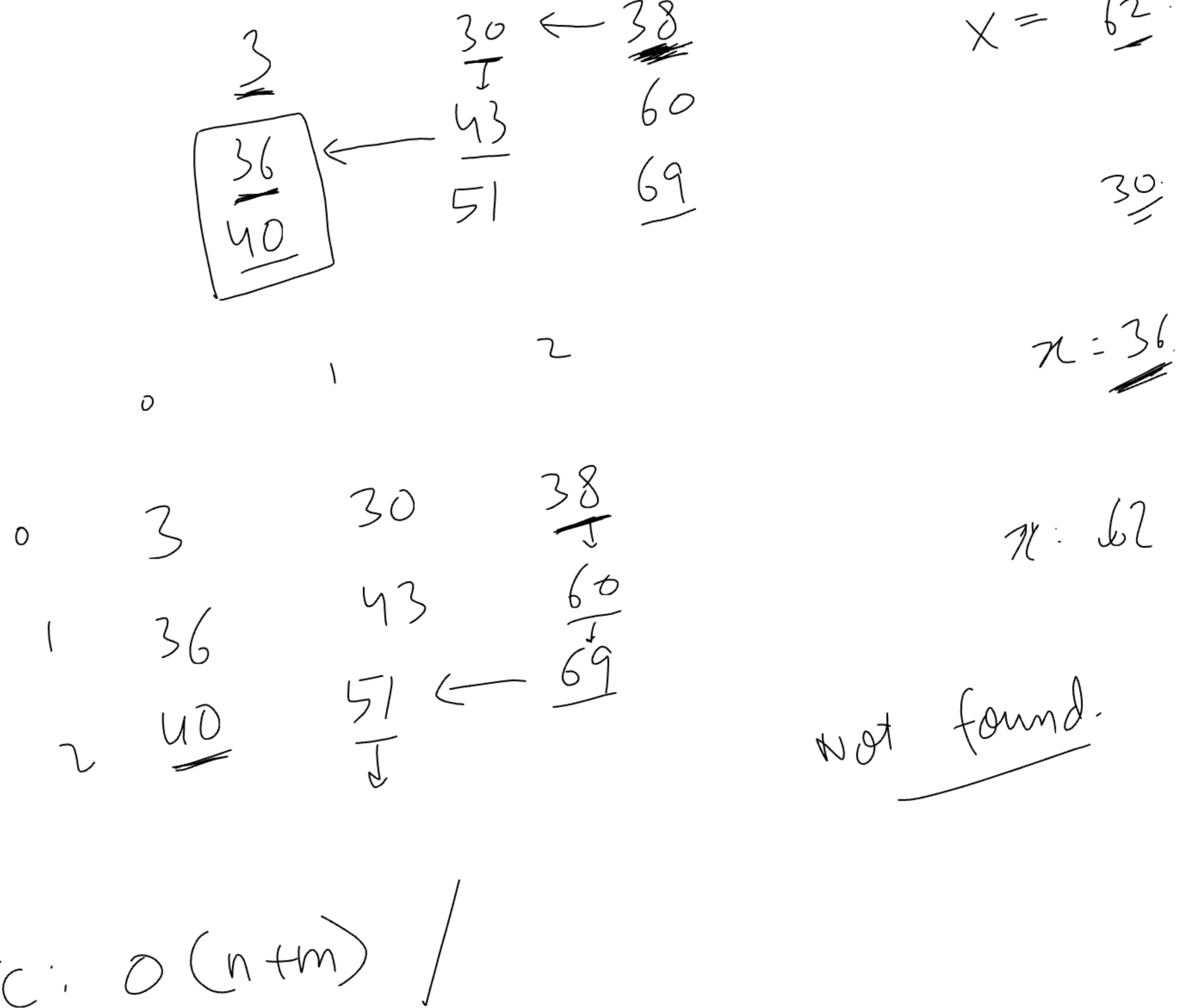


$n \times m$

BF: $O(nm)$

Row Binary Search: $O(n \log m)$

Col Binary Search: $O(m \log n)$



TC: $O(n+m)$ /
AS: $O(1)$

Unordered
(2, bye)
(1, hi)
(0, hello)

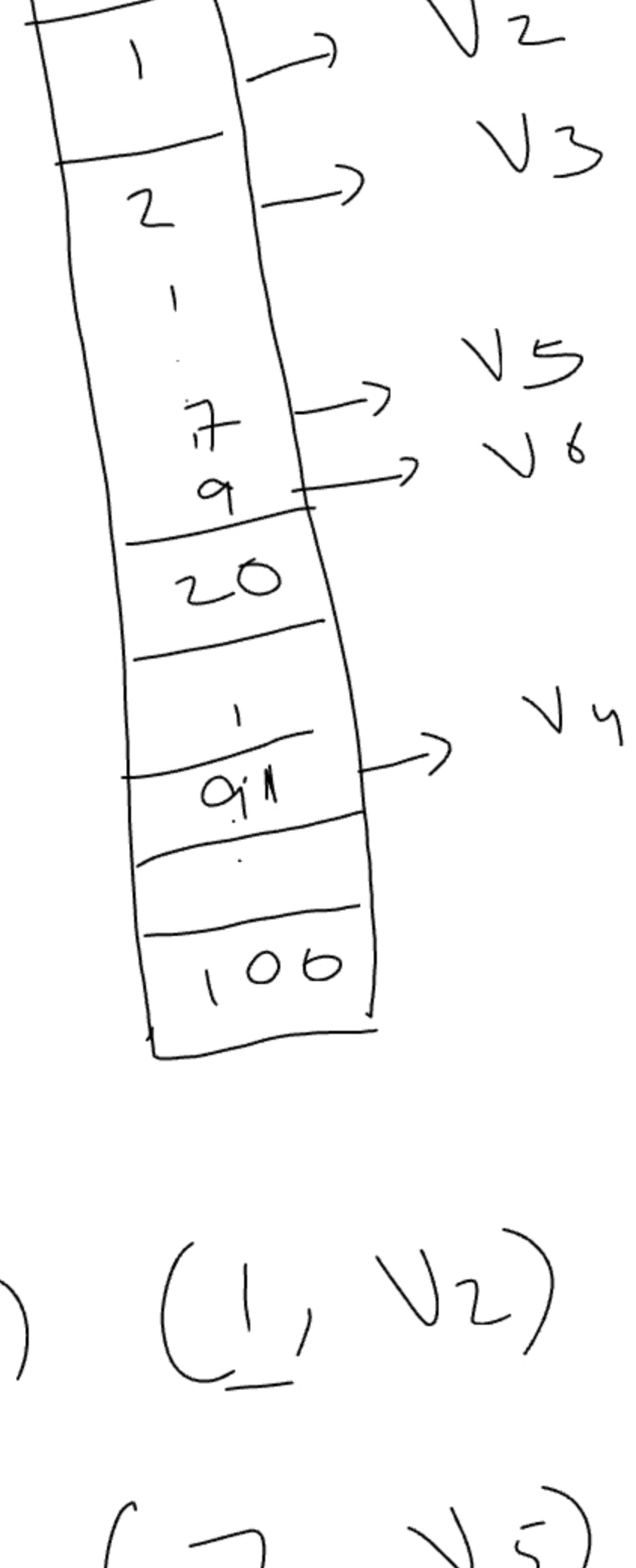
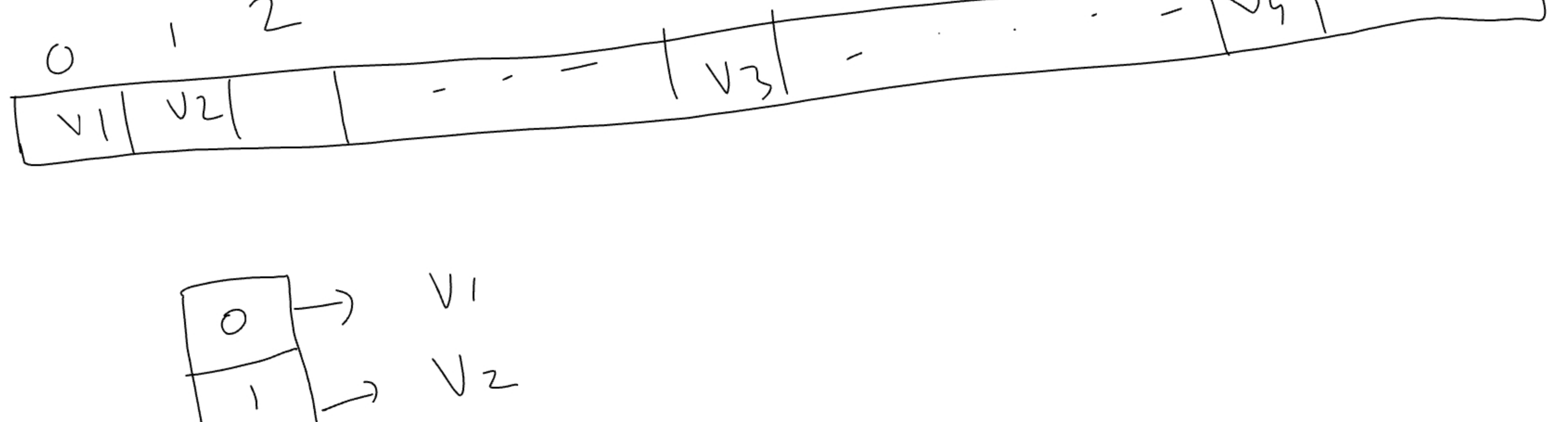
Ordered
(0, hello)
(1, hi)
(2, bye)

(key \rightarrow value)
 \downarrow

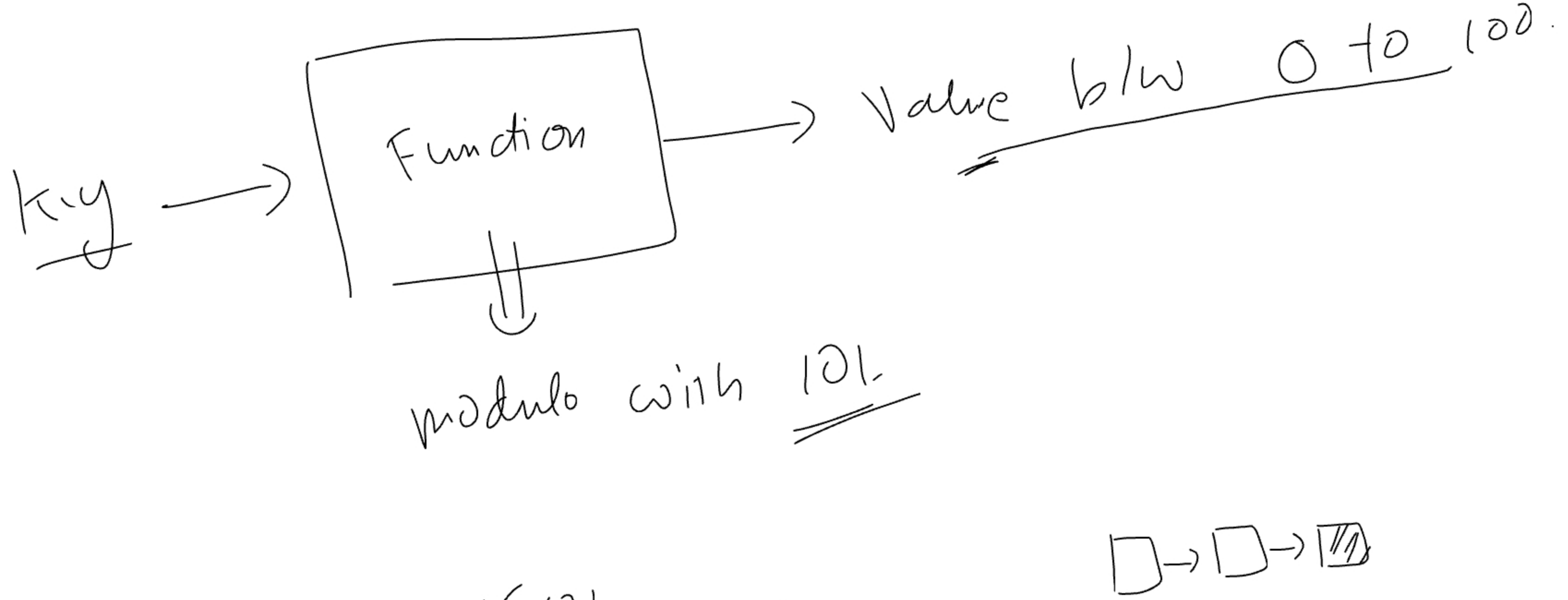
key value
0 \rightarrow hello
1 \rightarrow hi
2 \rightarrow bye

	<u>Unordered</u>	<u>Ordered</u>
Search:	$O(1)$	$O(\log n)$
Insert:	$O(1)$	$O(\log n)$
Delete:	$O(1)$	$O(\log n)$

(0, v1) (1, v2) (20, v3)
(100, v4) ...

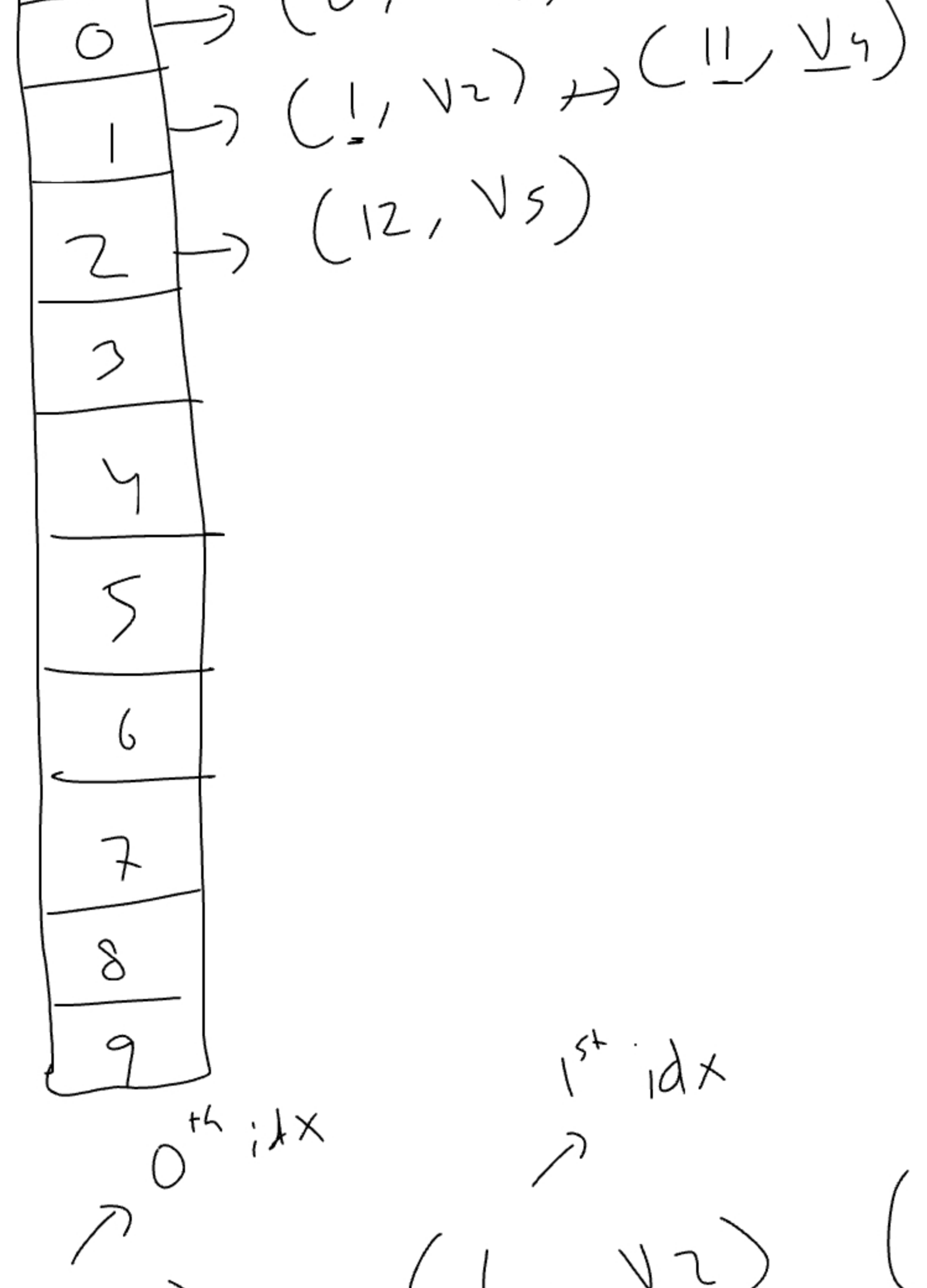


[(0, v1) (1, v2) (2, v3) ... (1000, v4)]
(7, v5) (9, v6) (107, v7)]
 \downarrow
(200 key-value pairs)



key1 = 0 $\xrightarrow{\% 101}$ 0
key2 = 101 $\xrightarrow{\% 101}$ 0

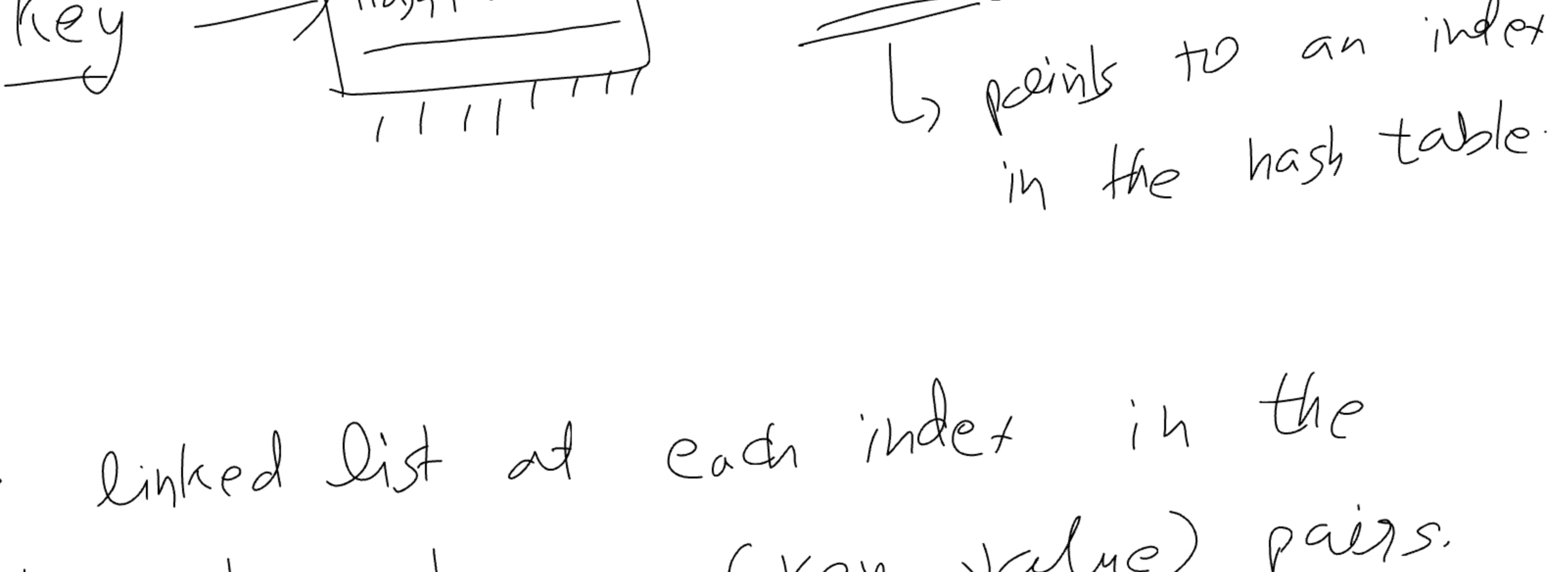
When 2 or more keys map to a single index in the hashtable, \rightarrow a collision has occurred.



Hash-Function:
 $\% 10$

[(0, v1) (1, v2) (10, v3) (11, v4)]
(12, v5)
 \downarrow 2nd idx

Hashtable internally are an array of linked lists.
The index of the array denotes the hashkey for the given keys.



* The linked list at each index in the hash table stores (key, value) pairs.

	<u>Hash Tables</u>	
	<u>Avg. case</u>	<u>Worst-case</u>
Insert	$O(1)$	$O(1)$
Delete	$O(1)$	$O(n)$
Search:	$O(1)$	$O(n)$

\times $\left(\begin{matrix} 0 & 1 & 2 & 3 \\ d & \leq & b & a \\ 3 \times 37^0 + 2 \times 37^1 + 1 \times 37^2 + 0 \times 37^3 \dots \end{matrix} \right)$