Hash Table = Data Structure

The	Am	I	Is	DIU

I want to search for the word "Is" from this String type array.

If my array size in "n", the time complexity for searching would be O(n)

We will store and retrieve words in a manner such that the time complexity is constant i.e O(1)

table → a function

Insert into Hash Table:
T[hash(String)] = String

Retrieve: String x = T[hash(String)]

Loops are not necessary for Hash functions

5 Element Hash Table: I want to insert the words The, Eth, We, Boy

Element Number	String
0	The
1	
2	
3	
4	

Eth (4+19+7)%5 = 0

Since both The and Eth have the same value for their hash function, there will be a **COLLISION**

Sample Hash Function: The (19+7+4)%5 = 30%5 = 0

Hash(String) =

T h e

Alternative Hash Function: (Instead of adding the value, concat them)

Hash(The) = 1974 % 5 = 4

Hash(hte) = 7194 % 5 = 4

- STILL THERE IS A COLLISION

P.T.O

Collisions are BAD. The more collisions there are, the more likely it is for Time Complexity to be Linear rather than Constant.

One of two techniques have to be used to RESOLVE COLLISIONS

- **Chaining Method**
- **Linear Probing**

Add

Search

Delete

Insert The, Boy, We, Oyb, Eth

Element Number	String
0	The
1	We
2	
3	
4	Boy

Eth →

Oyb →

Hash(The) = 0
Hash(Boy) = (1+14+24)%5 = 39%5= 4
Has(We) = (22+4)% 5 = 26 % 5 = 1
Hash(Oyb) = 4
Hash(Eth) = 39%5 = 4

CHAINING METHOD

0	1	2	3	4
The	We	Oyb	Eth	Boy

Insert The, Boy, We, Oyb, Eth

Hash(The) = 0
Hash(Boy) = 4
Hash(We) = 1
Hash(Oyb) = 4
Hash(Eth) = 0

Linear Probing

Value:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Hash function: KILL = (11+9+12+12) % 11 = 0. Use the Value box to find the values of each character. Use LINEAR PROBING FOR COLLISION RESOLUTION.

Hash Table:

0	1	2	3	4	5	6	7	8	9	10
~~Kill~~	KK		ZIA	KZK	AZ	AB	LIKE			EXAM

Linear Probing

SL NO	Operations	Collision =Yes/No	Hash Function Value	Actual index of the array where the operation is done Or false if invalid/not found	No. of boxes Traversed
1.	Insert(Kill)	No	0	0	1
2.	Insert(KK)	Yes	0	1	2
3.	Search(EHL)	No	3	Not found	1
4.	Insert(ZIA)	No	3	3	1
5.	Insert(AZ)	No	5	5	1
6.	Delete(KVK)	Yes	0	Not found	2
7.	Insert(KZK)	No	4	4	1
8.	Insert(AB)	Yes	3	6	4
9.	Insert(EXAM)	No	10	10	1
10.	Delete(Kill)	No	0	0	1
11.	Insert(LIKE)	Yes	4	7	4

SL NO	Operations	Collision =Yes/No	Hash Function Value	Actual index of the array where the operation is done Or false if invalid/not found	No. of boxes Traversed
1.	Insert(Kill)	No	0	0	1
2.	Insert(KK)	Yes	0	0	2
3.	Search(EHL)	No	3	Not found	1
4.	Insert(ZIA)	No	3	3	1
5.	Insert(AZ)	No	5	5	1
6.	Delete(KVK)	Yes	0	Not found	2
7.	Insert(KZK)	No	4	4	1
8.	Insert(AB)	Yes	3	3	2
9.	Insert(EXAM)	No	10	10	1
10.	Delete(Kill)	No	0	0	1
11.	Insert(LIKE)	Yes	4	4	2

Hash function: KILL = (11+9+12+12) % 11 = 0. Use the Value box to find the values of each character. Use LINEAR PROBING FOR CHAINING RESOLUTION.

Value:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Hash Table:

0	1	2	3	4	5	6	7	8	9	10

CHAINING

Value:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Hash Table:

0	1	2	3	4	5	6	7	8	9	10
KK			ZIA	KZK	AZ					EXAM