Jhymani Joseph
Part B: Simulation and Diagram Exercises

Exercise 1: Open Hashing (Separate Chaining)

Given the following set of keys:
5,22,17,18,35,101,16,0,8

hash table size = 10
Division Hash function = $h(k) = k \, mod \, 10$.

Keys	$h(k) = k \, mod \, 10$	Bucket Index
5	$5 \, mod \, 10 = 5$	5
22	$22 \, mod \, 10 = 2$	2
17	$17 \, mod \, 10 = 7$	7
18	$18 \, mod \, 10 = 8$	8
35	$35 \, mod \, 10 = 5$	5 (collision)
101	$101 \, mod \, 10 = 1$	1
16	$16 \, mod \, 10 = 6$	6
0	$0 \, mod \, 10 = 0$	0
8	$8 \, mod \, 10 = 8$	8 (collision)

Collisions occur when multiple elements map to the same bucket:

Bucket 5: Contains 5, 35
Bucket 8: Contains 18, 8

Collision Resolution method = Separate Chaining

Hash Table

Index	Bucket
0	0
1	101
2	22
3	
4	
5	5 → 35
6	16
7	17
8	18→ 8
9	

Exercise 2: Closed Hashing (Open Addressing)

Using the same set of keys:
5,22,17,18,35,101,16,0,8

hash table size = 10
Division Hash function = $h(k) = k \ mod \ 10$.

Keys	$h(k) = k \ mod \ 10$	Initial Bucket Index	Linear Probing Bucket Index
5	$5 \ mod \ 10 = 5$	5	5
22	$22 \ mod \ 10 = 2$	2	2
17	$17 \ mod \ 10 = 7$	7	7
18	$18 \ mod \ 10 = 8$	8	8
35	$35 \ mod \ 10 = 5$	5 (collision)	6
101	$101 \ mod \ 10 = 1$	1	1
16	$16 \ mod \ 10 = 6$	6 (collision)	9
0	$0 \ mod \ 10 = 0$	0	0
8	$8 \ mod \ 10 = 8$	8 (collision)	3

Collisions occur when multiple elements map to the same bucket:

Bucket 5: collision at 5 and 35 → 35 is placed in Bucket 6
Bucket 6: Collision at 16 → 16 is placed in Bucket 9
Bucket 8: Collision at 18 and 8 → 8 is placed in Bucket 9 (occupied),so then moved to Bucket 3

Collision Resolution method = Linear Probing

Clustering happens when many keys get stored next to each other, making future insertions take longer. Value 35 was supposed to go into bucket 5, but it was already occupied by value 5, so it moved to bucket 6, which was the next available free index/bucket. Value 16 was supposed to go into bucket 6, but value 35 occupied that bucket, so value 16 moved to bucket 9, which was the next available free index/bucket. Lastly value 8 was supposed to go into bucket 8, but value 18 was in that spot, so it checked the next available bucket which was 9 but this was already taken by value 16, and so it did a final scan from index/bucket 0 and found the next available bucket was 3 and finally moved value 8 to bucket 3.

The more elements cluster together, the more time it takes to find an empty spot. Future insertions have to search through more occupied spots, making the table less efficient.

Hash Table

Index	Bucket
0	0
1	101
2	22
3	8
4	
5	5
6	35
7	17
8	18
9	16

Exercise 3: Impact of Poor Table Sizes

Consider the set of keys:
5, 10, 15, 20, 25, 30, 35, 40

hash table size = 10
Division Hash function = $h(k) = k \, mod \, 10$.

Key	Index
5	5
10	0
15	5 (collision)
20	0 (collision)
25	5 (collision)
30	0 (collision)
35	5 (collision)
40	0 (collision)

Collisions occur when multiple elements map to the same bucket:

Bucket 5: Contains 5, 15, 25, 35
Bucket 0: Contains 10, 20, 30, 40
Keys only go to **two buckets (0 and 5)**, causing **severe clustering** and many collisions.

Collision Resolution method = Separate Chaining

Hash Table

Index	Bucket
0	10 → 20 → 30 → 40
1	
2	
3	
4	
5	5 → 15 → 25 → 35
6	
7	
8	
9	

Consider the set of keys:
5, 10, 15, 20, 25, 30, 35, 40

hash table size = 11
Division Hash function = $h(k) = k \ mod\ 11$.

Key	Index
5	5
10	10
15	4
20	9
25	3
30	8
35	2
40	7

No collisions.

Hash Table

Index	Bucket
0	
1	
2	35
3	25
4	15
5	5
6	
7	40
8	30
9	20

Using a **prime number** for the table size helps distribute keys **more evenly**, reducing collisions and making lookups faster.

Exercise 4: Extra Credit

Consider the set of keys:
12, 23, 34, 45, 56, 67, 78, 89

hash table size = 11
Division Hash function = $h(k) = k \, mod \, 11$.

Key	$h(k) = k \, mod \, 11$	Quadratic Probes $(h\,(k) + i^2) \, mod \, 11$	Final Index
12	$12 \; mod \, 11 = 1$		1
23	$23 \; mod \, 11 = 1$	$(1 + 1^2) mod \, 11 = 2$	2
34	$34 \; mod \, 11 = 1$	$(1 + 2^2) mod \, 11 = 5$	5
45	$45 \; mod \, 11 = 1$	$(1 + 3^2) mod \, 11 = 10$	10
56	$56 \; mod \, 11 = 1$	$(1 + 4^2) mod \, 11 = 7$	7
67	$67 \; mod \, 11 = 1$	$(1 + 5^2) mod \, 11 = 2$	4
78	$78 \; mod \, 11 = 1$	$(1 + 6^2) mod \, 11 = 9$	9
89	$89 \; mod \, 11 = 1$	$(1 + 7^2) mod \, 11 = 3$	3

Quadratic Probing: If a collision occurs at index h(k), we try the next available slot using the formula:

$$h\,(k, i) = (h\,(k) + i^2) \; mod \; (table \; size)$$

where i = 1,2,3,… until an empty slot is found.

quadratic probes for each key:

- 12 → Initial hash 1 (no collision).
- 23 → Initial hash 1, probes 2 ($1 + 1^2$ mod 11).
- 34 → Initial hash 1, probes 2 (collision), then 5 ($1 + 2^2$ mod 11).

- 45 → Initial hash 1, probes 2 (collision), 5 (collision), then 10 ($1 + 3^2 \bmod 11$).
- 56 → Initial hash 1, probes 2, 5, 10 (all full), then 7 ($1 + 4^2 \bmod 11$).
- 67 → Initial hash 1, probes 2, 5, 10, 7 (all full), then 4 ($1 + 5^2 \bmod 11$).
- 78 → Initial hash 1, probes 2, 5, 10, 7, 4 (all full), then 9 ($1 + 6^2 \bmod 11$).
- 89 → Initial hash 1, probes 2, 5, 10, 7, 4, 9 (all full), then 3 ($1 + 7^2 \bmod 11$).

Each key follows the quadratic pattern $(h + i^2)\ mod\ 11$ until an open slot is found.

Collision Resolution method = Quadratic Probing

Hash Table

Index	Bucket
0	
1	12
2	23
3	89
4	67
5	34
6	
7	56
8	
9	78
10	45

Quadratic probing spreads out keys effectively, reducing clustering and preventing long chains of occupied slots. Unlike linear probing, it avoids filling consecutive positions, improving distribution. However, keys with the same initial hash follow the same probe sequence, causing some secondary clustering.