

Assignment 1 (CO1)

1) Two applications of sorting in data structures.

Ans Sorting is the process of arranging data in a specific order (Ascending or descending).

- Database Query Optimization → Sorting allows efficient searching and merging.
 - Ex → Sorting student records by Roll no which helps in faster binary search.

- Data Analysis → Sorting is used to find medians, rankings and duplicate detection.

Ex → in a E-commerce site, products are sorted by price or rating.

2)

Static hashing vs Dynamic hashing

Ans • Static hashing → The hash table size is fixed if records increase collision rises.

Ex → Student_ID stored in a fixed table.

- Dynamic hashing → The hash table expands/shrinks dynamically depending on data growth.

Ex → Extendible hashing in databases.

3)

Internal vs External Sorting

Ans Internal sorting → data is stored entirely in main memory (RAM). Suitable for small datasets.

Ex → Quicksort on an array of 1000 elements.

- External sorting → used when data is stored in external storage devices like hard disk.

fit in memory, sorting is done using both memory and disk.

Ex -> External merge sort for sorting $\log B / \log$.

4) Buffer management in external sort

Ans • In External Sorting, disk access is expensive.
• Buffer management allocates blocks of memory as I/O buffers instead of reading/writing one record at a time large chunks are moved reducing I/O operations.

5) External sorting with example

Ans. External sorting is used when data is larger than available memory.

1) Divide large file into chunks (runs).

2) Sort each run in-memory.

3) merge runs until final sorted file is obtained.

Ex -> Storing 2TB log file in BigQuery

6) Advantage of Double hashing over Linear Probing

Ans • Linear Probing: Collisions are resolved by searching next slots sequentially, which causes primary clustering.

• Double hashing uses a second hash function to calculate probe step size, reducing clustering and spreading keys uniformly.

Ex -> $H_1(k) = k \bmod 7$, $H_2(k) = 1 + (k \bmod 5)$.

f) Folding method

Ans Table size = 10.

- Key 987654321 \rightarrow $9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 = 45 \rightarrow 45 \mod 10 = 5$
- Key 5643231 \rightarrow $5 + 6 + 4 + 3 + 2 + 3 + 1 = 24 \rightarrow 24 \mod 10 = 4$
- Key 3478654 \rightarrow $3 + 4 + 7 + 8 + 6 + 5 + 4 = 39 \rightarrow 39 \mod 10 = 9$

g) Double Hashing.

Ans $H_1(k) = k \mod 13$, $H_2(k) = 12 - (k \mod 12)$.
 $H(i) = (H_1(k) + i * H_2(k)) \mod 13$.

i	1	2	3	4	5	6	7	8	9	10	11	12
44	41			18	32	59	73	22	31			

$$18 \mod 13 = 5$$

$$44 \mod 13 = 5$$

$$22 \mod 13 = 9 \quad (22 - (22 \mod 12) = 22 - 8 = 9)$$

$$44 \mod 13 = 5 \rightarrow (5 + 1 \times 12) \mod 13 = 17$$

$$59 \mod 13 = 7 \quad \left\{ \begin{array}{l} i=1 \rightarrow (5 + 1 \times 12) \mod 13 = 17 \\ i=2 \rightarrow (5 + 2 \times 12) \mod 13 = 29 \end{array} \right.$$

$$32 \mod 13 = 6 \quad \left\{ \begin{array}{l} i=1 \rightarrow (5 + 1 \times 12) \mod 13 = 17 \\ i=2 \rightarrow (5 + 2 \times 12) \mod 13 = 29 \end{array} \right.$$

$$31 \mod 13 = 5 \rightarrow \left\{ \begin{array}{l} i=1 \rightarrow (5 + 1 \times 12) \mod 13 = 17 \\ i=2 \rightarrow (5 + 2 \times 12) \mod 13 = 29 \end{array} \right.$$

$$73 \mod 13 = 8$$

h) Open Addressing Example.

Ans $H(k) = k \mod 6$, $k = 24, 19, 32, 44, 57$.

* 24 mod 6

1	1	1	2			
24	19	32	44	57		

$$24 \bmod 6 = 0$$

$$19 \bmod 6 = 1$$

$$32 \bmod 6 = 2$$

$$44 \bmod 6 = 2 \rightarrow \text{Next free} = 3$$

$$57 \bmod 6 = 3 \rightarrow \text{Next free} = 4$$

10) mid-square Method

Ans $K = 625$, Table size = 100

$$625^2 = 390625$$

$$\text{mid-digits} = 06$$

$$06 \bmod 100 = 6 \quad (\text{1st} \rightarrow \text{6}), \quad (\text{2nd} \rightarrow 06) = (06) \bmod 100$$

$$\text{Hash value} (= 6_{\text{base}} + i + 1_{\text{base}}) = (06) \bmod 100$$

11) K-way merge ($k=3$)

Ans $k=3$ means merging 3 runs simultaneously.

Ex ->

$$\text{Run 1} = [3, 7, 10], \text{ Run 2} = [1, 4, 8], \text{ Run 3} = [2, 5, 9]$$

Process : Pick Smaller $\rightarrow [1, 2, 3, 4, 5, 7, 8, 9, 10]$.

12) Linear probing - Example

Ans Table size = 7, $h(k) = k \bmod 7$: (1) base (F)

$$K = 50, 700, 76, 85, 92$$

$$50 \bmod 7 = 1$$

$$700 \bmod 7 = 0$$

$$76 \bmod 7 = 6$$

$$85 \bmod 7 = 1 \rightarrow \text{Collision} = 2$$

1	2	3	4	5	6
700	50	85	92		76

$$92 \bmod 7 = 1 \rightarrow \text{collision} = 3$$

13) Comparison of hashing techniques.

Ans i) Mid-Square \rightarrow Spreads keys well but Depends on mid number

ii) Folding \rightarrow Breaks keys into parts and adds ; good for long no

iii) Data Analysis \rightarrow Selects specific digits ; useful if distribution is uniform.

iv) Double Hashing \rightarrow Best for reducing clustering, gives uniform distribution.

Ques

14) External merge sort example.

Ans Run 1 = [3, 7, 10]

Run 2 = [1, 4, 8]

Run 3 = [2, 5, 9]

Smallest \rightarrow [1, 2, 3] \Rightarrow

next small \rightarrow [4, 5, 7]

next = [8, 9, 10]

K = [1, 2, 3, 4, 5, 7, 8, 9, 10].

15) Time complexity of external sort.

Ans formula = $O(M \log k (N/M))$, where

$N = 7000$, $M = 700$, block = 100 \rightarrow $k = 700/100 =$

No. runs = $N/\text{block} = 7000/100 = 70$

passes = $\log_7 (70) \approx 2$.

Time complexity $\approx O(7000 \times 2) = O(14,000)$.