HASHING :

Agenda :

↳ Problem (Try to optimize)
↳ Hashing

problem : 1 hr 40 mins

Problem :

→ You are given a list of size N.
→ This list has some numbers. Range [1 - 1000]
→ Ex : [1, 10, 78, 999, 1000, 1]
→ You will get Q - queries. This is also a list of size Q.
→ for every query, you have to print True if that query x in list else False

Example :

lst = [4, 5, 80, 999, 1000, 2]

query = [80, 500, 899, 1]

Ans : True False False False

Approach 1

★ **Linear Search**

⇒ No. of linear search = Q (no. of queries)
⇒ Time Complexity of Linear Search = O(N)

★ Total operations = Q × N

⇒ N = 100
⇒ Q = 100

★ Total operatn = 100 × 100
= 10^4 operations

10^8 operations = 10 sec
10^7 operations = 1 sec

→ 10^8 = 10 s
→ 1 operatn = $\frac{10}{10^8}$ sec
→ 10^7 operatn = $\frac{10}{10^8} \times 10^7$ = $\frac{10}{10}$
= 1 sec

$\Rightarrow$ Time for 10^4 queries = 1 ms

$N = 10^6$
$Q = 10^6$

Total operatn = $10^6 \times 10^6 = 10^{12}$

time = $\frac{10}{10^8} \times 10^{12}$

= 10×10^4 sec

= 10^5 sec

= 1.15 days

★ Approach 2

→ Sort the list and apply binary search for all queries.

→ Binary Search = $O(\log N)$

→ Sorting = $N(\log N)$

→ Total operations = $N(\log N) + Q \times \log(N)$

(N: size of list; Q: size of query list)

⇒ $N = 10^6$

⇒ $Q = 10^6$

\# Total time $= 10^6 (\log 10^6) + 10^6 (\log 10^6)$

$= 2 \times 10^6 (\log 10^6)$ → $\log 10^6 \approx 20$

$= 2 \times 20 \times 10^6$

$= 4 \times 10^7$

$= 4 \times 1$ sec

$= 4$ sec

★ Approach 3:

↳ Create a new list of size → 1001

↳ Indexes : 0 → 1000

★ 9'll fill all values as False in new list.

★ list = [0, 1, 2, --- , 1000, 2, 500]

Ex : no. => 0, 9

list = [0, 5, 2, 3, 8, 8, 2, 5]

query = [1, 5, 9, 8, 8, 5, 8, 5]

new = [T F T T F T F F T F]

0 1 2 3 4 5 6 7 8 9

⇒ for i in query :

if new[i] == True :

else : False

★ Total operations :

new[i] ⇒ O(1)

= Generate of flag list + Answering queries

flag list → T/F → N operation

Answering queries → [x → new[x] ⇒ O(1)] → Q times

⇒ N + Q

# N = 10^6
Q = 10^6

Total time = 2×10^6
= 2×0.1
= 0.2 sec

Range = 0 - 1000
Range = 1 - 10^{10}

inefficient in terms of space.
space consumed : multiple Gbs

★ Approach 4

Sets → unique values
adding → O(1)
Searching → O(1)

} Hashing

Add all elements in a set
Search for your query in set.

Adding data in Set:
⇒ 1 → O(1)
⇒ N → O(N) # worst case

Searching data:
⇒ 1 → O(1)
⇒ Q → O(Q)

$N = 10^6$

$Q = 10^6$

Total operations $= N + Q$

$= 10^6 + 10^6$

$= 2 \times 10^6$

Total time $= 2 \times 0.1 = 0.2$ sec

---------->

Space : At max size $\Rightarrow \simeq N$

$\simeq N$

$\Rightarrow 10^6 \simeq 1$ mb