3.4 - Hashing-1

Wednesday, November 26, 2025 9:07 PM

Two Data structures based on Hashing in Java

→ Hash Map

→ Hash Set.

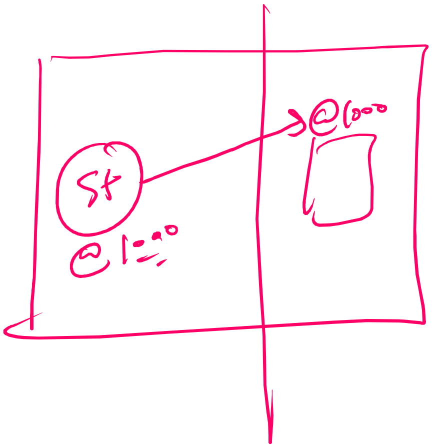
Hash Set

↓

Set is a collection of unique elements.

S = {10, 2, 4, 3} ↙ 4

→ $\underbrace{\text{HashSet<Integer> st}}_{\text{Stack}} = \underbrace{\text{new HashSet<>()}}_{\text{heap.}};$



→ .add(x) : adds the element to the set.

→ .size() : returns the size of the set.

→ .remove(x); removes element x from the set.

→ isEmpty(); returns true if set is empty otherwise

→ isEmpty(): returns true if set is empty ...
false.

Types of Set in Java

- → Hashset: Elements are randomly stored in a Hashset.
- → Tree set: Elements are stored in sorted order.
- → Linked HashSet: Maintains the order of insertion.

Complexity Analysis.

	add( )	contains( )	remove( )
			add( )
				remove( )

	C	R	U	D
HashSet	O(1)	O(1)	O(1)	O(1)
Tree Set	O(logn)	O(logn)	O(logn)	O(logn)

self balancing BSTs

Generic Class

HashSet < Integer>

——— < Double>

< Boolean>

______ < Double

______ < Person>

______ < Node>

______ < Elephant>

→ generic class is a class which can be replaced by any valid class during the definition of the object.

HashMap

→ set of key-value pairs.

key	value
Jaiswal →	13
Sundar →	40
Rahul →	10
Kohli →	40

→ key in a hash map are always unique

→ values can be duplicate.

→ HashMap<String, Integer> mp = new HashMap<>();

→ All the different operations in a HashMap happen on keys.

Integer Integer.
roll No. marks

Hash map

- put (k,v) → o(1)
- get (k) → o(1)
- size() → o(1)
- remove(k) → o(1)

Types of maps

→ HashMap : stores keys in random order

→ Tree Map : stores keys in sorted order o(log n)

→ linked Hash map : maintains the order of insertion of keys

⇒ mp.get (key) → null ← null pointer exception..

We never directly do a .get(k).
We first check if the key is present or not
.containsKey (k);

Iterating a set and map

[for each]