

Hashing

V-X Hashact :-* contains all unique elements & removes duplicate. elements in the set are not ordered (jumbled) The Complexity: - Insut/Add > O(1); Search/contain > O(1);

Delete/remove > O(Delete/remove > O(1); Array (OLL) BST (OCH) Hashlet < Integer > set = now Hashlet < >(); - same of array wit Set. add(1); Set. add (2); set add (1); -> no need (won't wensider duplicate) Set. contains (1) // true Iterator: " import iterator first hasNext next # does'nt guarentee order of items when iterested will be same next time. We case: for iteration in set Mashet Iterator it = Set: iterator(); < while (ithounext ()) 2 print (it. next ()); Hashmap

At Must have key, value pair where key should be unique

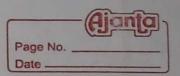
(etring Theger? map = new HashMap <7(); HashMap < String, Integer > map = new HashMap < 7();

key value

(wuntries) (population) * Insuling ... map. put (" India", 70); map. put (" USA", 90); map. put ("China", 70); print (map); -> 0/p

{china=70, US=90, India=70} order not

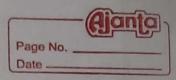
guarenteed.



* updating - new value with same key name : map.put("china", 200); (value only) Note: in Updating -> exist: updates value of key respectively -> not expt: adds new key, value pair. Jearching (key) > map. contains key ("China"); //true * Get (value) -> map. get ("China"); //200 from key map. get ("Indonesia"); //null Two types of for doop: - 1. int[] arr = 212,15,184;
for (int i=0; i < arr. length; i+)2 y Print (arr [i] + " "); 2. for (int val: arr)?

print (val + ""); * for (map. Entry < string. Integer > e: map. entry set (3) } print (e. get key ()); print (e. get Value ()); 2. keyset > Set < String > keys = map. keyset (); for (string key; keys) { print (key+ " " + map.get (rey)); gives value * Deletion > map. semoue ("china"); print (map); ~ Implementation (in next Page) V.V. imp for interviews

*	Implementation search
14	functions > put, get, contains key, remove, size, reglet
	all in OCL) complexities
V	Implemented as array of LL.
	the second to the second of th
V+	* basic Declaration: - import java uti. *;
	public clars Hash Code 2
generics (detatype is dependable) Static class Hashmap < K, V> {
	check private class Node f
	datatype og key is K _ K key; 4 value is V
	4 value is V L V value;
	dependable on public Node (K key, V val) { (A) Node (K key, V val) { (A) Node (K key, V val) {
	CO.O. T. ADICUIC
	y this, value = val;
	J
	no. of nodes inside buckete private unt n;
	bucket. length=N = private unt N; "(int[] arr) = private Lineallyt <node> [] buckets</node>
-	
7	to findex are buckets)
	ander 0 > 0 0 0 0 Aine nodes are stored
	modes in linked list
Due to gener	rice, Java mein type nhi define (a) Suppressivarnings ("Unche 140 d")
Kouro chall	ga but not in L.L.:und public Hashmap () {
	initialized bucket size with 4 < this. N = 4;
"we broin"	mitialised butlet with LL size 4 - this buckets = new Linked List [4]
	some indices of bl. 2 this buckets [i] = new new Linked List <>()
4	laga dige



PUT T.C = 0(2) 2 worst=0(m) - act the bucket index "bi" i.e. index of buckets array. " " data " "di" ic at bi jo L.L. stored hai usra index in need bi and key to get ai. If di gives -1 :. key doesn't exist : add new node at that bi and update the size of in? - Else if key exist .. update value at di. noturn type = void, ang = k key, vreil int bi = hashfunction (key); int di = searchin LL (key, bi); if (di == -1) { // key does nt exist -> add new node + increase n buckets[bi]. add (new Node (key, val)); property of LL yntt; else & 11 key exists -> get the value then place new one Node data = buckets [bi]. get (di); data, value = val; Just to make sure ex hi index < double lambda = (double) m/N; if (lambda 72.0)? (let K=2)

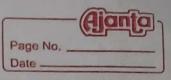
Liwritten like threshold

rehash (); this : double
(must) + bada LL banta na chala jaye MashFunction -> return type = Jut, ourg = K key. unt bi = key hash (ode (); return Mathales (bi) % N; · howhlode generates unique code, here it will generate based on key

(cauge to to like 12765 or - 314411)

· We got bi positive values only, that too in range of N.

```
* securchinLL > return type = Int, arg = k rey, bi
 provide LL inside bi > Linked List< Node> el = buchets [bi];
                       for (i=0 to Ul. size 1) Top
                  if ( al. get(i), key == key) {
                           , return i; > 'di' found
                       return - 1; -> di does not exist
* rehash function -> return type = void, no arep
   I firstly jo buchets me tha use old buchet of daalo
      buchets ka lise double karo
     saare naye buchets & lineal list bouras
   i aur har old bucket ke index pe jo Il hai une get kan
       aur put car do roga waha pe vid wall ant
  O linked list < Node > Gold Bucket = buckets;
  >2 buchets = now Linked Unt[N*2];
  >3) for (i=0 to buchets, length or N#2) {
            buchetaci] = now Linkedlist <> ();
        for (i=0 to old Buchet. length) ?
            linked Hot < Node> Il = Oblowlite(17)
               for (j=0 do el. size()){
                Node node= le.get (j);
                 put (node, key, node, value);
 main fn: - psurne - HashMay String, Integer > map = new Hashmaper
                         generics will take the datatypes.
        map.put ("India", 190); --- more put
         : Arraylist (string > keys = map keyset ();
            for (i=0 to keys. size()){
print ( Keyp. gelli) + " " + map. get ( keyp. get (i))) is
map. remo ue ( "India");
             prive ( map. get (" India")); - null.
```



vetum type = 'V", ang = K key GET int bi = hashfunction (rey); int di = searchinel (key, bi); 1 (di==-1) } elsed ye ex node has acche se socho get the node from - Node node = bucheti[bi]. get(di); y return node value; -> get the value. 42 if lay exists return type = boolean, ang = k key Cowains key but br = v Int di= ~ ele & return true; riewing purposes return type=V, arg = K key. 1 de Remove if (di==-1) { return null; } > = DNE 1, novemone Node node = bucheta (bi); remove (di); return node value; - viewing purpous. boolean, no ourg 1s Empoty return n==0; -> if no nodes in hashmap thew Empty keyset return = Array LIST < K> , no ang Anaylot < k> rep = new Amaylot <> (); v.easy just Labko add krdo key List for (i=0 to buckets, length)[understand conceptually LL < Node> ll = buchets[i]; me our return kardo. for (j=0 to Il-size) Node node = legetli); y y keys, add (node, key)

return keys;