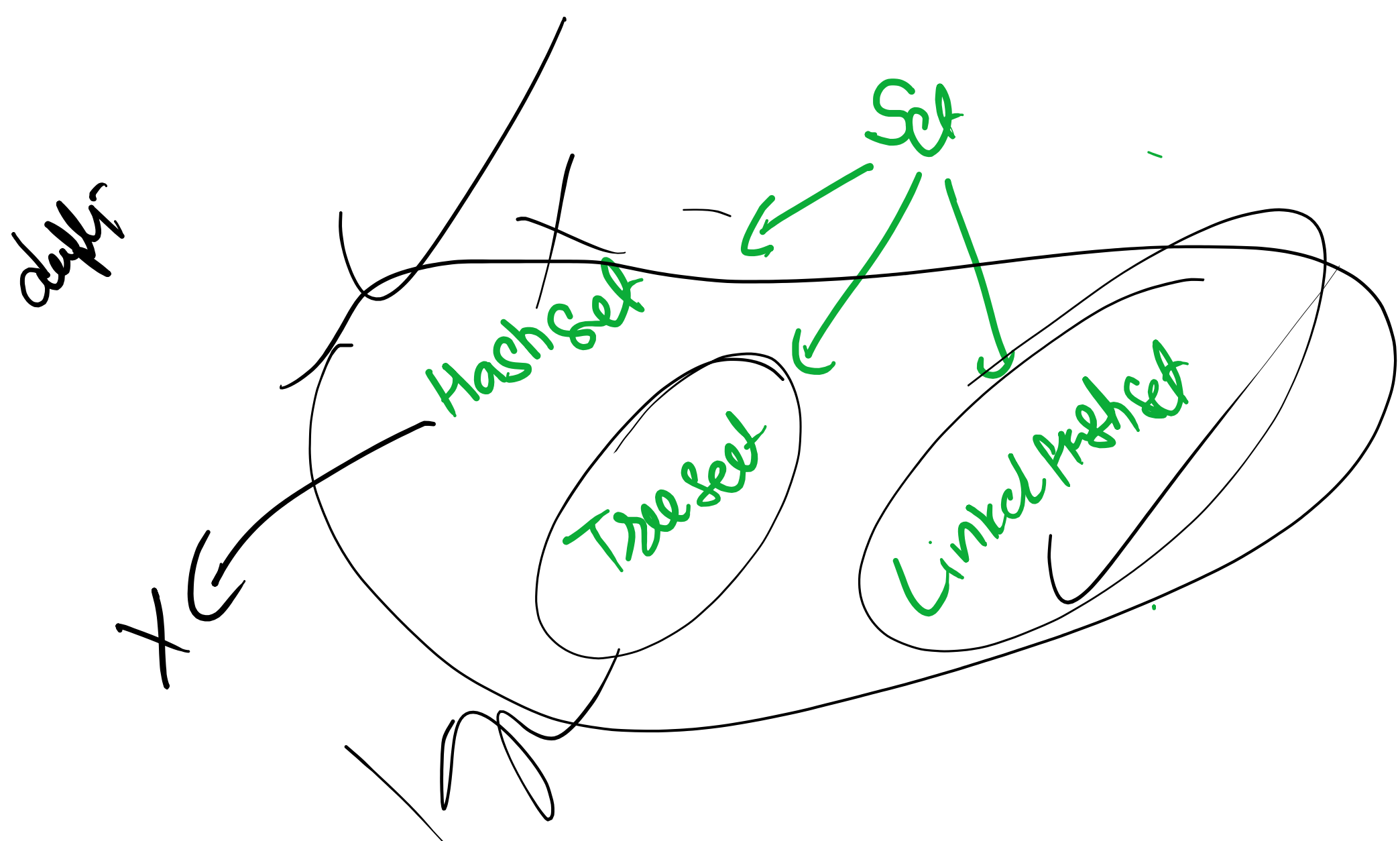
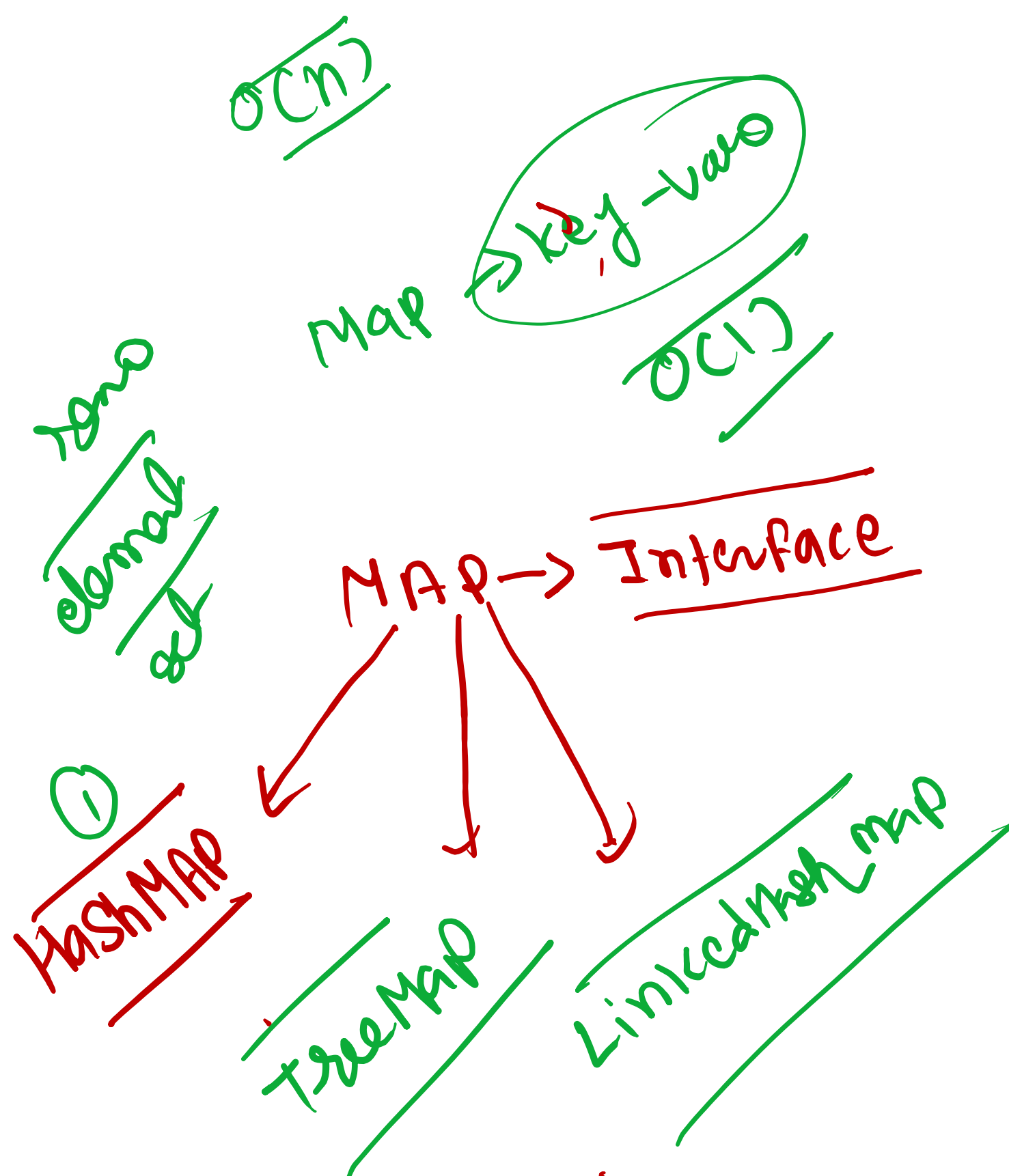


Map | Set

(i) Map \rightarrow key-value
name \rightarrow marks

(A 95)	(B 99)	(C 75)	(D 68)
-----------	-----------	-----------	-----------



Given two integer arrays `nums1` and `nums2`, return an array of their *intersection*. Each element in the result must appear as many times as it shows in both arrays and you may return the result in **any order**.

Handwritten example of two arrays and their intersection:

Array 1: [2, 3, 1, 4, 5, 3, 7, 1, 2, 4, 5]
Array 2: [2, 4, 3, 3, 4, 3, 7, 2, 2, 7, 11, 15, 3]
Intersection: [2, 3, 4, 5, 7]

Int	Int
2	1
3	1
1	1
5	2
7	1
4	1

```
for (int i = 0; i < arr2.length; i++) {  
    if (map.containsKey(arr2[i])) {  
        ll.add(arr2[i]);  
        map.put(arr2[i], map.get(arr2[i]) - 1);  
    }  
}
```

Handwritten result: [2, 1, 3, 3, 4]

Handwritten calculations:

- Sort
- 7 8 9
- 10
- 1 2 3 4 5

Handwritten code for finding the intersection of two arrays:

```
for(int key : map.keySet()) {  
    if(map.get(key)) {  
        int c = 0;  
        while (map.containsKey(key)) {  
            c++;  
            key++;  
        }  
        ans = Math.max(ans, c);  
    }  
}
```

Int	Boolean
2	F
5	F
7	T
4	F
1	T
8	F
9	F
15	T
13	T
16	F
3	F

Handwritten notes: $val \rightarrow val-1$, $val+1$