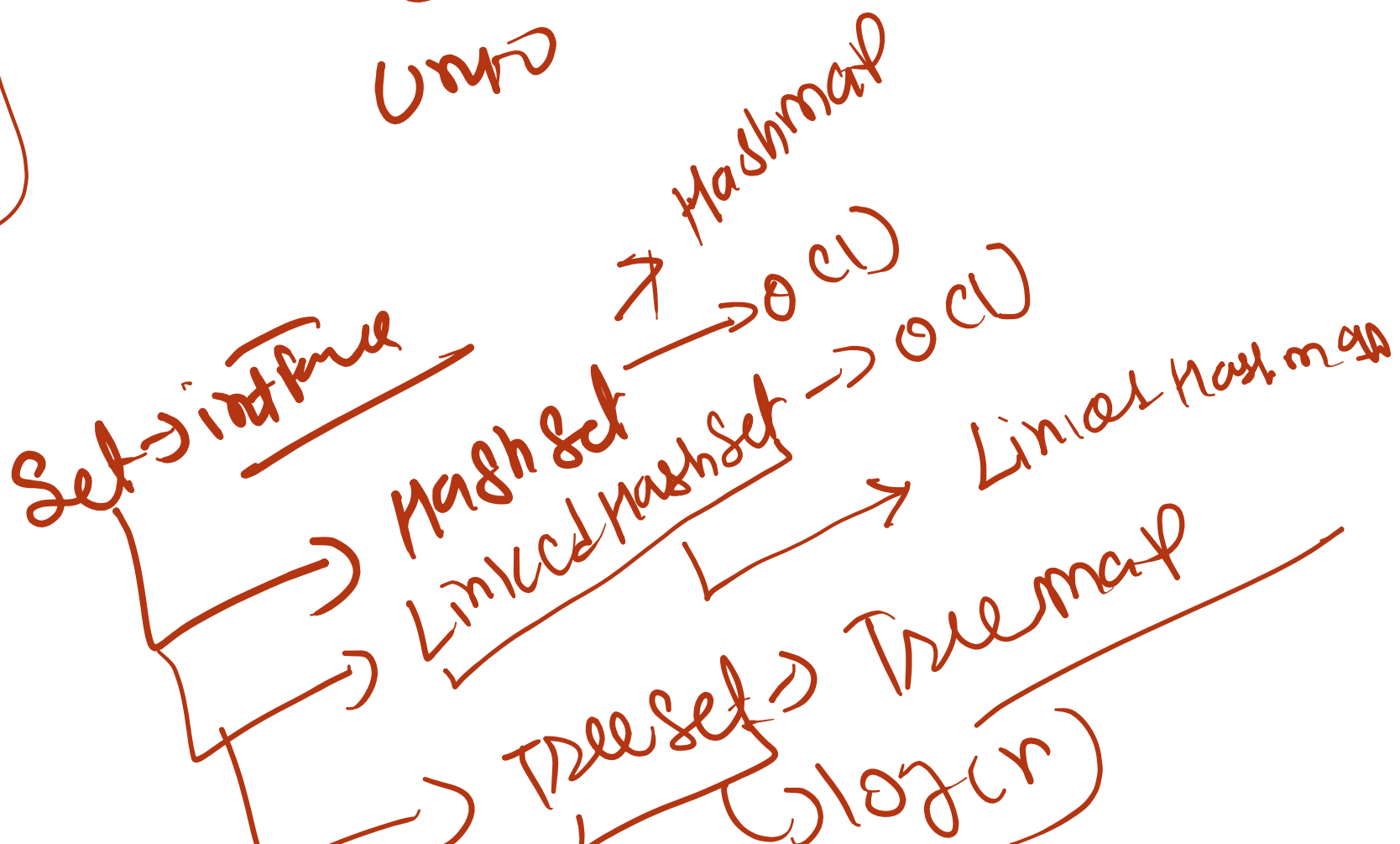
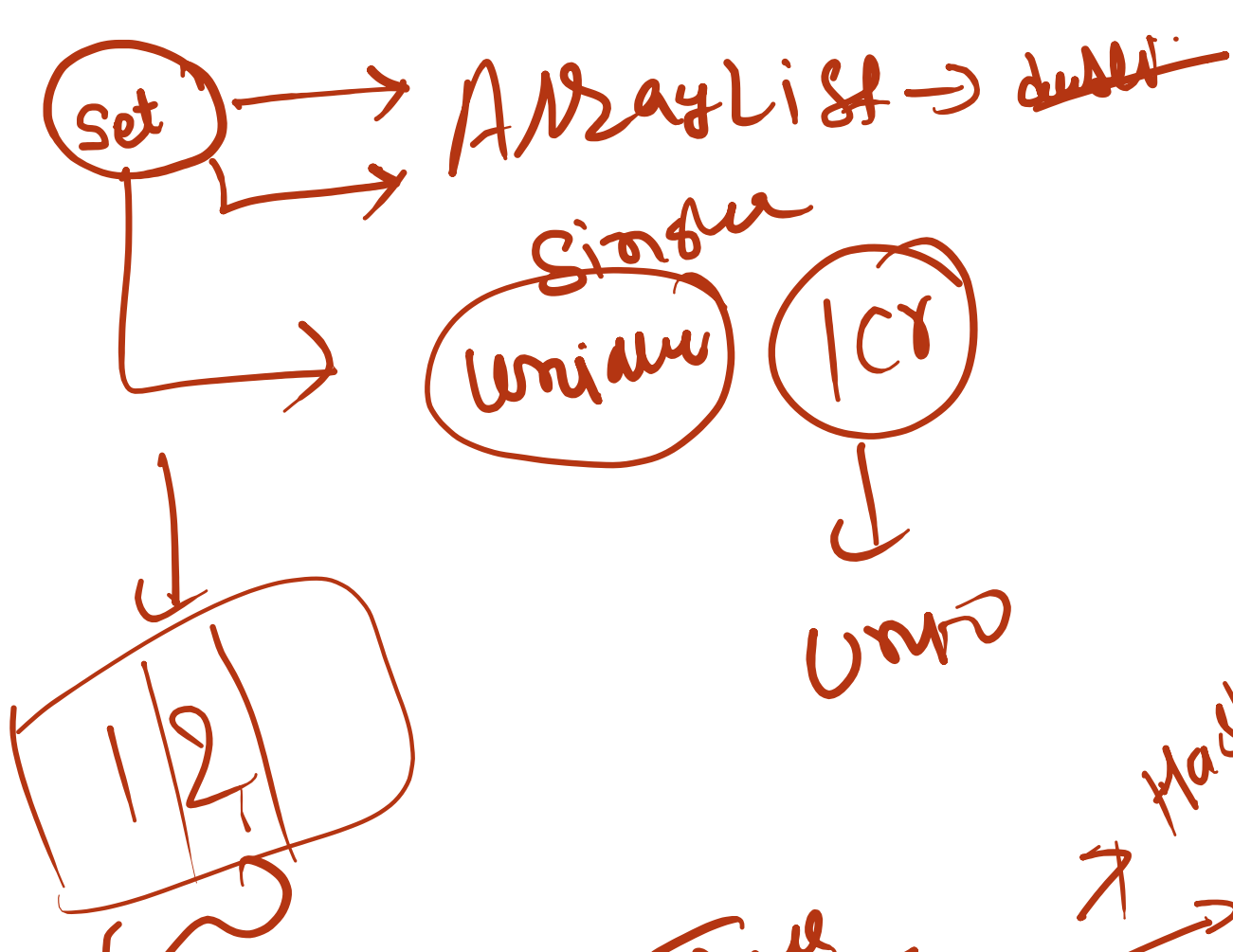


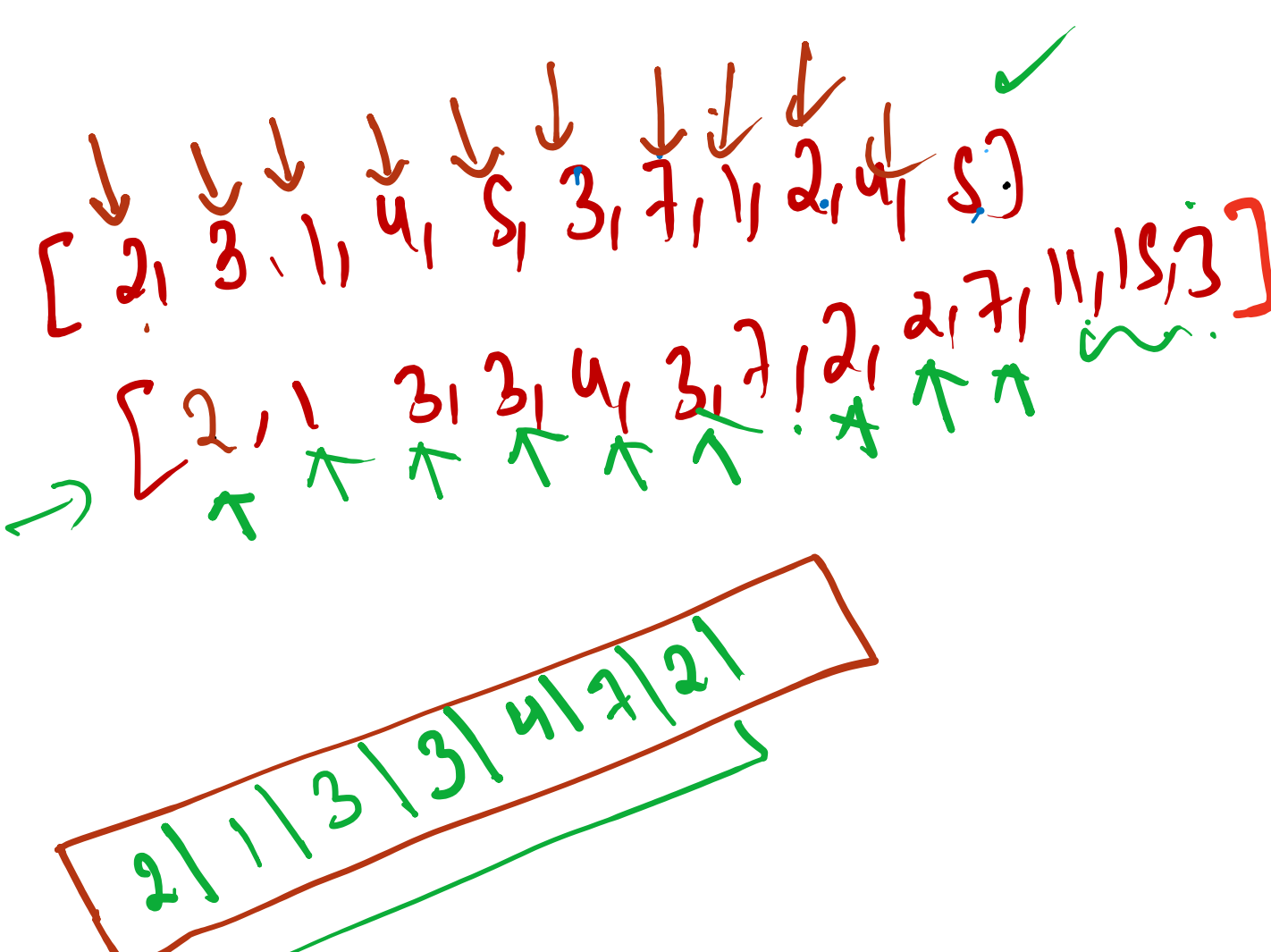
Map Set heap

1/2/2

hash



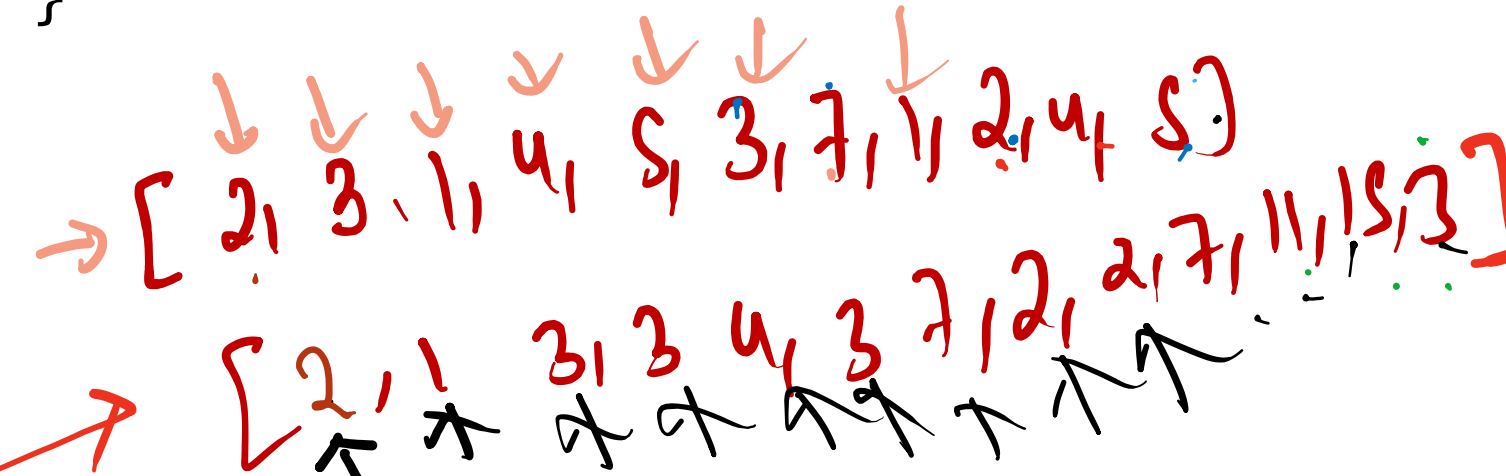
Given two integer arrays `nums1` and `nums2`, return an array of their intersection. Each element in the result must appear as many times as it shows in both arrays and you may return the result in any order.



2	0
3	0
1	1
4	1
5	2
7	0

```

public static int[] Intersection(int[] arr1, int[] arr2) {
    HashMap<Integer, Integer> map = new HashMap<>();
    for (int i = 0; i < arr1.length; i++) {
        if (map.containsKey(arr1[i])) {
            map.put(arr1[i], map.get(arr1[i]) + 1);
        } else {
            map.put(arr1[i], 1);
        }
    }
}
    
```



```

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

6x

max = 5

2	0
3	0
1	1
4	1
5	2
7	0

4, 5, 7, 2, 1, 8, 9, 15, 13, 16, 3, 2, 1, 5

1	2	3	4	5	7	8	9	15	13	16
---	---	---	---	---	---	---	---	----	----	----

5

1/2/2

7

1, 2, 3, 4, 5

15, 16

13

4, 5, 7, 2, 1, 8, 9, 15, 13, 16, 3, 2, 1, 5

ans = 5

```

for (int i = 0; i < arr.length; i++) {
    if (set.contains(arr[i] - 1)) {
        int c = 0;
        int v = arr[i];
        while (set.contains(v)) {
            c++;
            v++;
        }
        ans = Math.max(ans, c);
    }
}
    
```

1	2	3	4	5	7	8	9	15	13	16
---	---	---	---	---	---	---	---	----	----	----

1, 2, 3, 4, 5

1, 2, 3, 4, 5