Hash Map

key, value
↓
unique

Rohini	→ 12
Noida	→ 15
Gurugram	→ 20
Janakpuri	→ 34
Dwarka	→ 17

Syntax

```
HashMap<String, Integer> map = new HashMap<>();
```

functions → O(1)

add — `map.put("Rohini", 12);` `map.put("Noida", 17);`

remove → O(1) — `map.remove("Noida");`

Contains key → O(1) — `map.containsKey("Noida")` → true / → false

size → O(1) — `map.size();`

value get → O(1) — `map.get("Rohini") = 12`

print — `System.out.println(map);`

O(1) → exact
→ average

HashSet

HashSet< Integer> set = new HashSet<>();

add set.add(s)

remove set.remove(s)

contains set.contains(7) → true / false

size set.size();

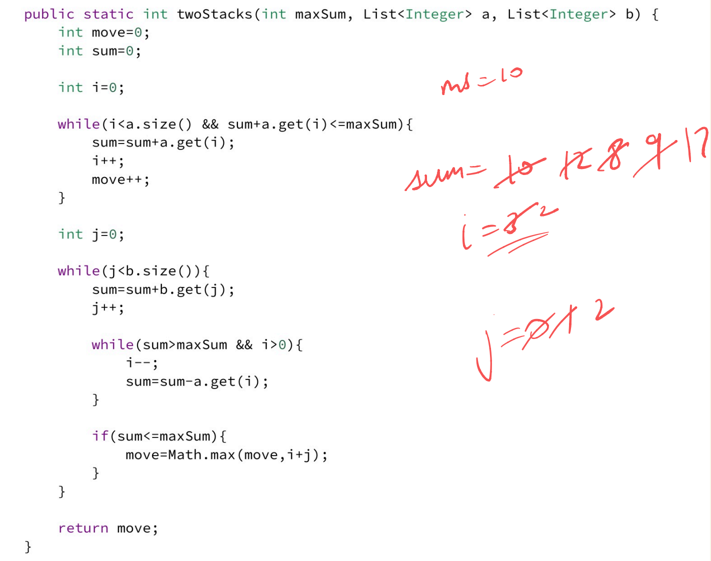
{set.add(s)
set.add(s)

set.size();

1 , , , 5

(1, 2, 4, 5)

10



```
public static int twoStacks(int maxSum, List<Integer> a, List<Integer> b) {
    int move=0;
    int sum=0;

    int i=0;

    while(i<a.size() && sum+a.get(i)<=maxSum){
        sum=sum+a.get(i);
        i++;
        move++;
    }

    int j=0;

    while(j<b.size()){
        sum=sum+b.get(j);
        j++;

        while(sum>maxSum && i>0){
            i--;
            sum=sum-a.get(i);
        }

        if(sum<=maxSum){
            move=Math.max(move,i+j);
        }
    }

    return move;
}
```

ms = 10

sum = ~~10~~ ~~12~~ ~~8~~ ~~4~~ 17

i = ~~3~~ 2

j = ~~0~~ ~~1~~ 2

0	4	0	2
1	2	1	1
2	4	2	8
3	6	3	5
4	1		