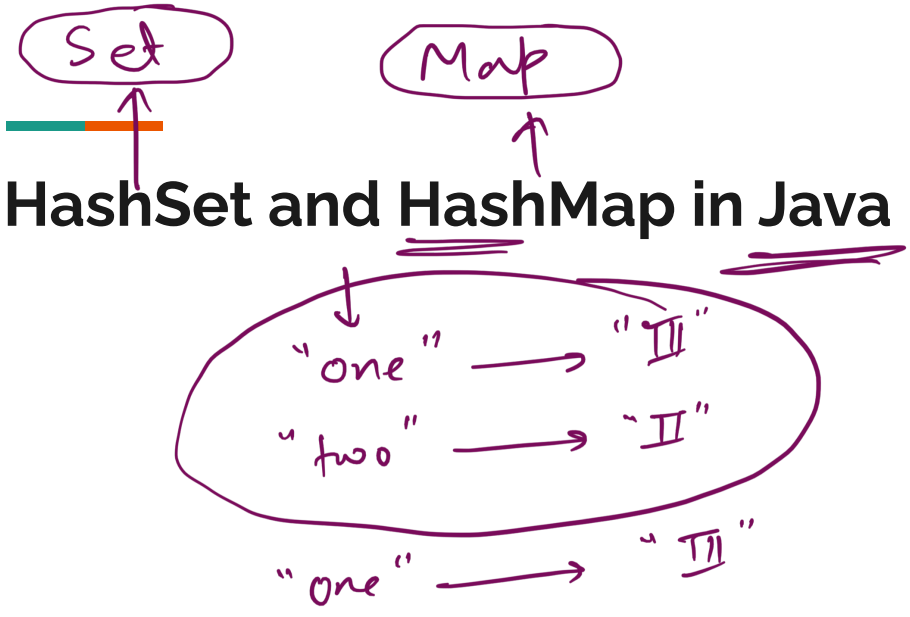
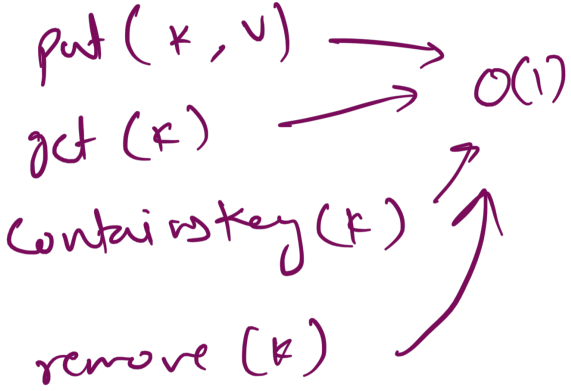
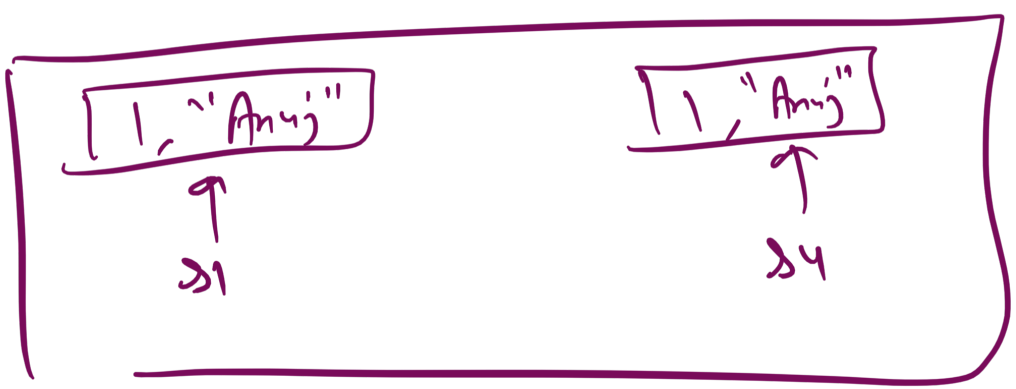
Hashing & HashMap Basics





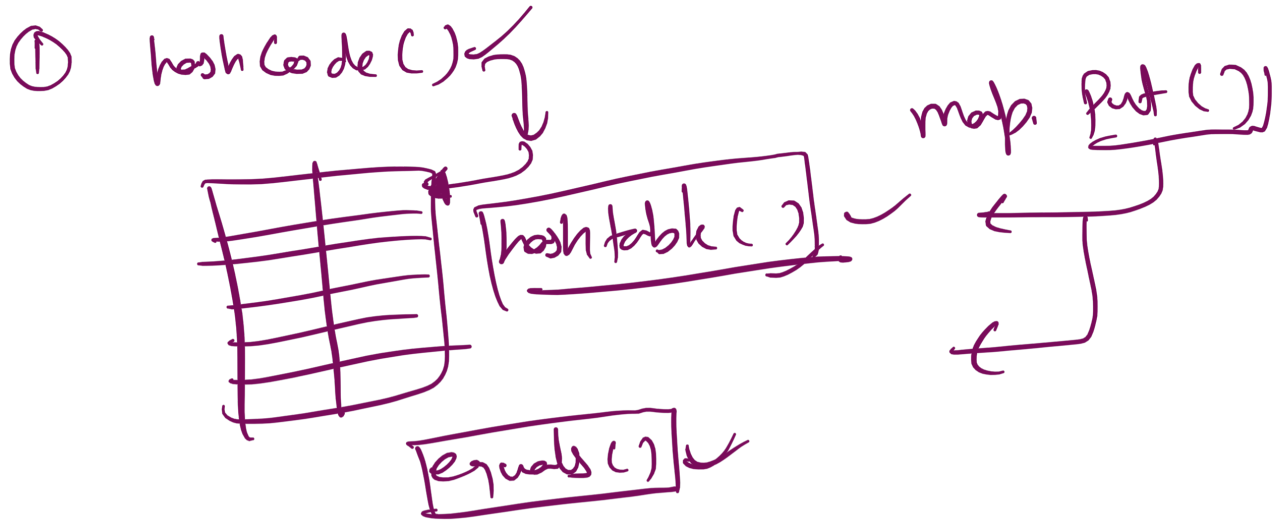
Internal working of HashMap



Shallow check
↳ references

Set

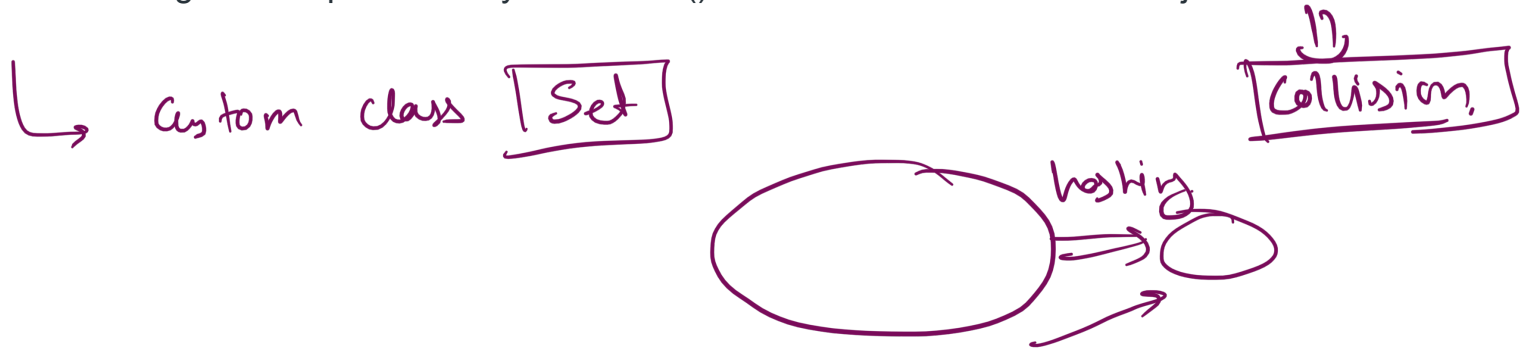
hashCode &
equals.



Map implement

hashCode and equals contract

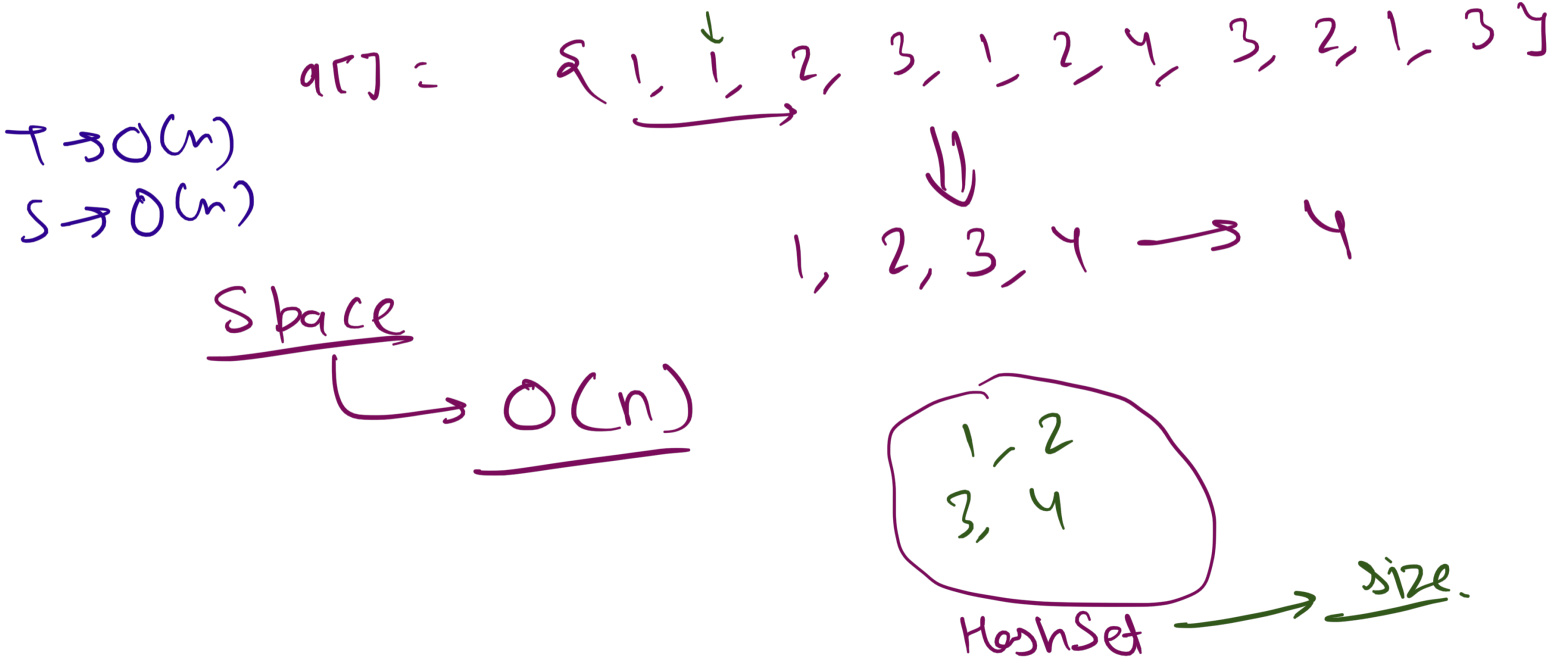
- During the execution of the application, if hashCode() is invoked more than once on the same Object then it must consistently return the same Integer value
- If two Objects are equal, according to the equals(Object) method, then hashCode() method must produce the same Integer on each of the two Objects.
- If two Objects are unequal, according to the equals(Object) method, It is not necessary the Integer value produced by hashCode() method on each of the two Objects will be distinct.



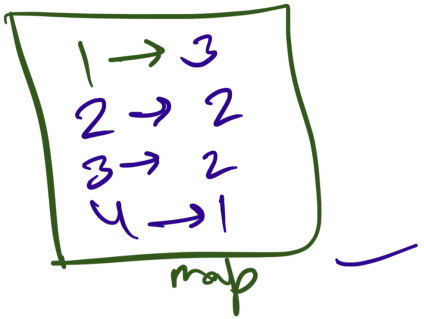
1 = 2 then $h(1) = h(2)$

if 1 ≠ 3 then $h(1) = h(3)$

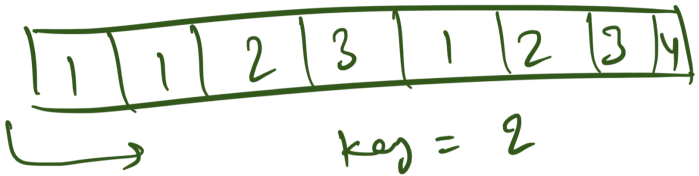
Count Distinct Elements



Frequency of Elements in an Array



T → O(n)
S → O(n)



1 →
2 →
3 →
4 →

Q queries.

→ O(Q.n) → O(Q) + O(n)

Preprocessing

Pair with given sum in an Unsorted Array

int a[] = {1, 3, 3, 1, 4, 5, 2, -1, 4}

sum = 8.

HashSet

O(n) → T

O(n) → S

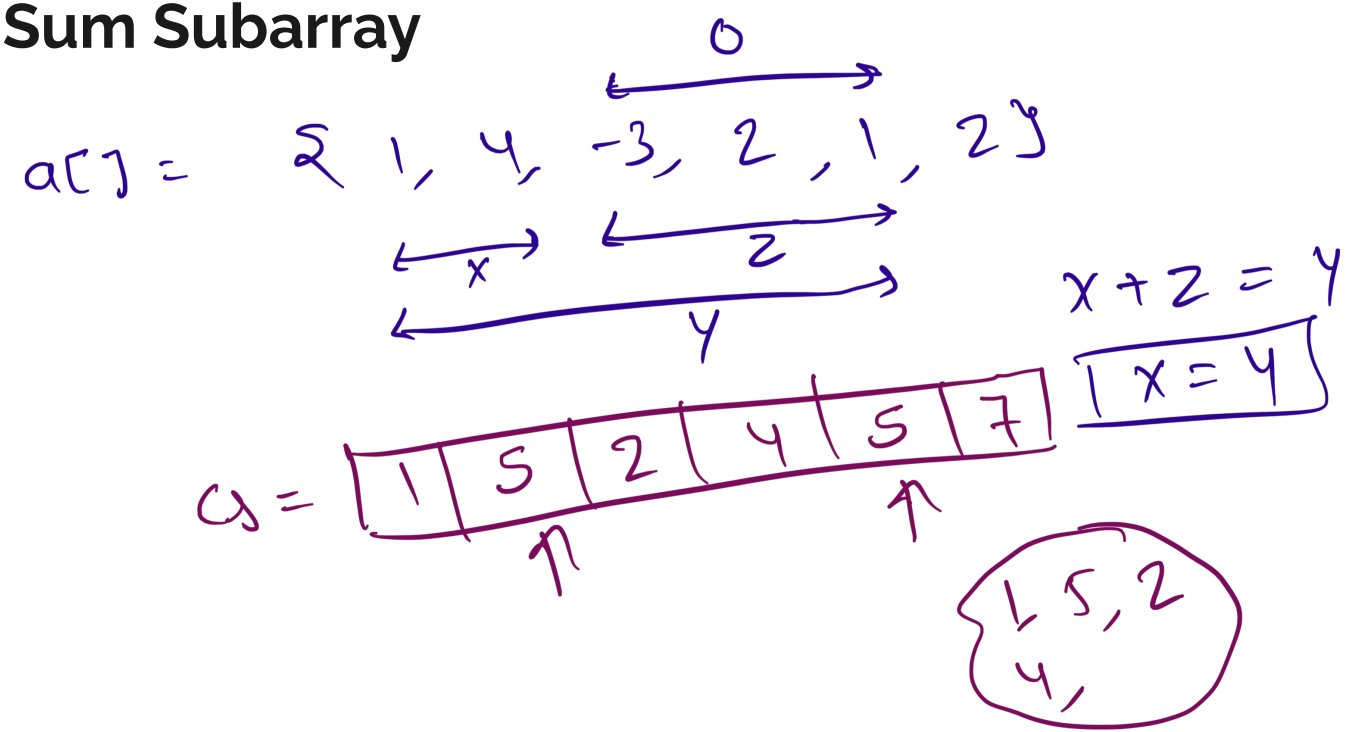
x → element

complimentary

(sum − x) + x

= sum

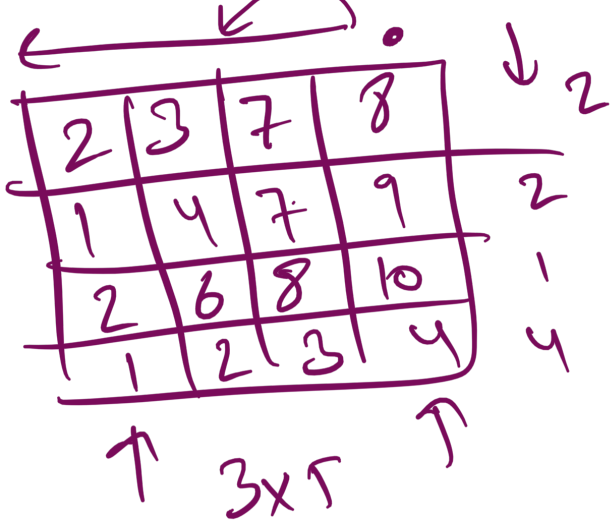
Zero Sum Subarray



Practice Problems

1. Subarray with given Sum.
2. Intersection of Two Arrays.
3. Union of Two Arrays.
4. Find the largest subarray with zero sum.
5. Count distinct elements in every window of size k.

N x M → odd



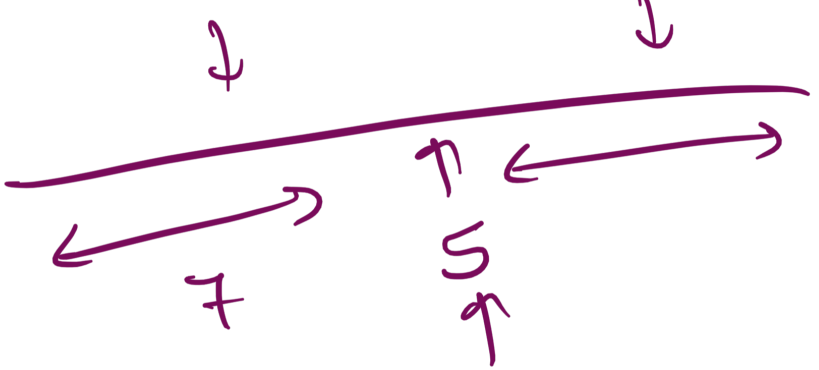
l = 1

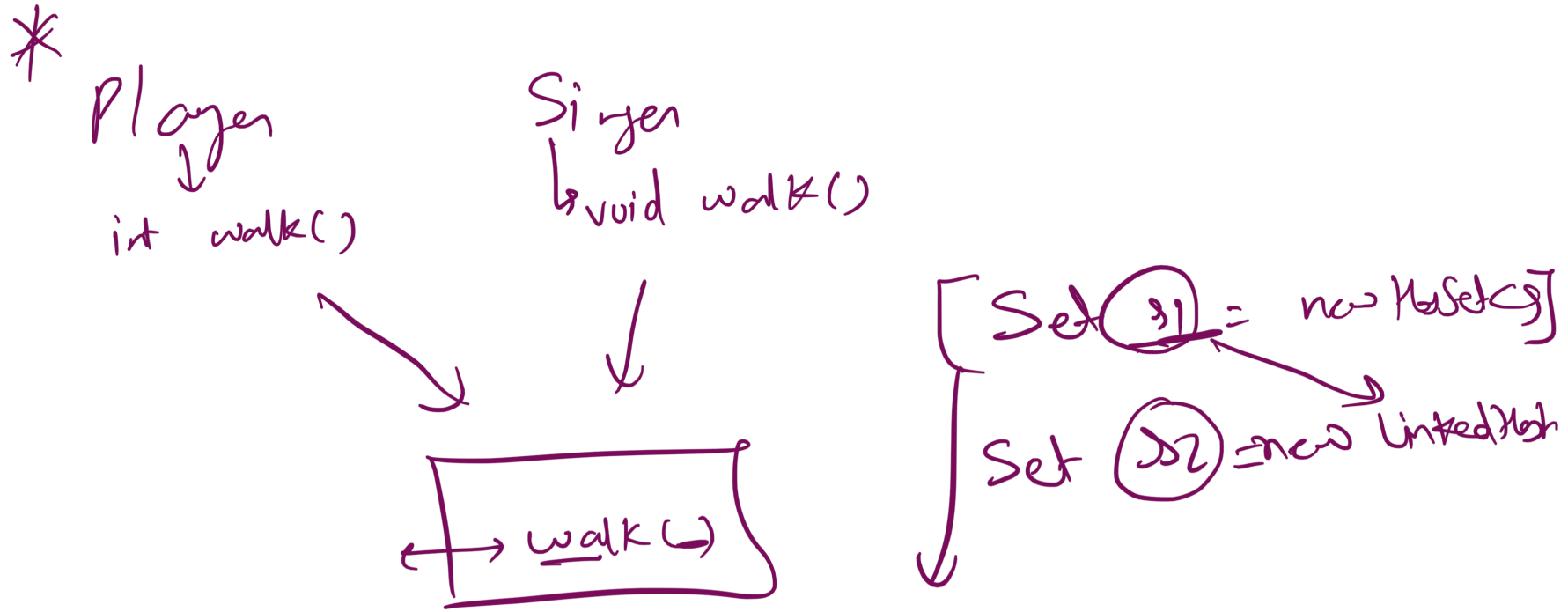
r = 10

mid = 5

r2 S-1 = 4

total = 15




Player
int walk()
Singer
void walk()
walk()
Set s1 = new HashSet<>
Set s2 = new LinkedHash