PW SKILLS

Maps & Sets

Introduction to HashSets

Interface/Data Structure

size, insert, remove, search → O(1)

In HashSet, occurence of every element is 1.

STL and important methods in Hashsets

- add()
- size()
- contains()
- remove()
- toArray()

How to iterate in Hashset

for each loop

Why not for/while loop ? → Because there is no concept of index

Ques:

Q : Count Number of Distinct Integers After Reverse Operations

arr = 12, 24, 36, 41, 21, 42, 63, 14

ans = 8

arr = 13, 24, 31, 12, 31, 42, 13, 21

ans = 6

arr = 2, 2, 2, 2, 2, 2

ans = 1

[Leetcode 2442]

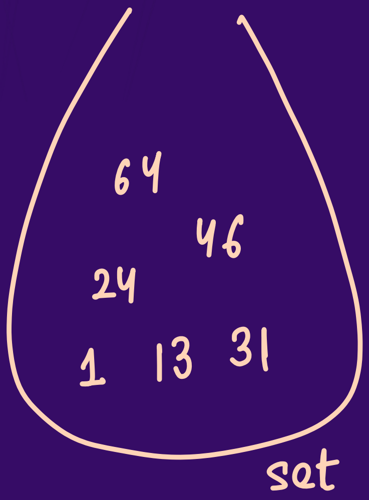
Ques:

Q : Count Number of Distinct Integers After Reverse Operations

arr = { 1, 13, 24, 31, 46 }

T.C. = $O(n)$

S.C. = $O(n)$



return set.size()

[Leetcode 2442]

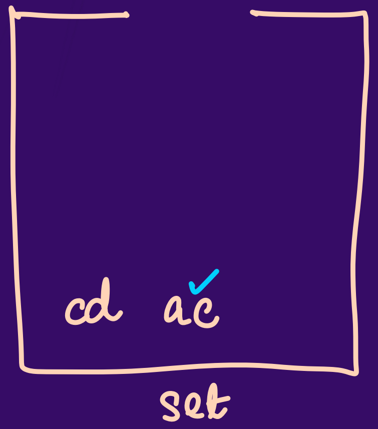
Ques:

Q : Find Maximum Number of String Pairs

arr = { cd, ac, dc, zt, ca, tu }

dc ca cd tz ac ut

count = ~~0~~ ~~1~~ 2



[Leetcode 2744]

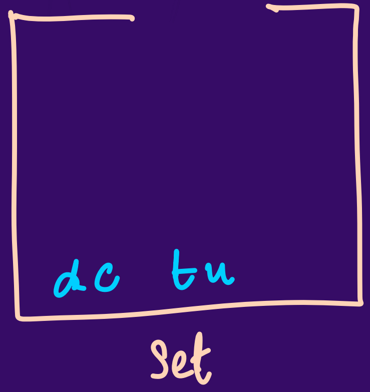
Ques:

Q : Find Maximum Number of String Pairs

arr = { cd, ac, dc, ca, dc, tu }

3 pairs ✗

2 pair



Count = $\cancel{0}\,\cancel{1}\,2$

T.C. = $O(n*l)$

S.C. = $O(n*l)$

where 'l' is avg. length of string

[Leetcode 2744]

Introduction to HashMaps

↓

data structure

insert, remove, search → $O(1)$

put

pairs → key, value

In a HashMap, there can be 2 or more keys with same values.

But keys are unique.

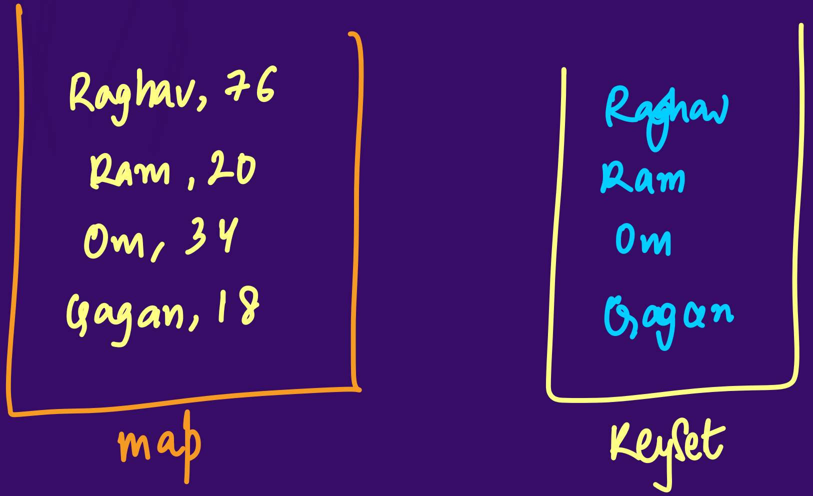
STL and important methods in maps

- ✓ put() insertion of (key, value) pair / update
- ✓ get() map.get(key) → gives the value of that key
- ✓ size()
- ~~find()~~
- ✓ remove() removes the pair → remove(key)
- ✓ containsKey() search for key
- ✓ countainsValue() search for value
- entrySet()
- ✓ Key Set ()

How to iterate in Hashmap

↓

for each loop , map.keySet()



Raghav, 76
Ram , 20
Om, 34
Gagan, 18

map

Raghav
Ram
Om
Gagan

keySet

Ques:

Q : Valid Anagram

s = "ate"   t = "eat"

s = "Raghav"   t = "avangR"

Brute Force

↓

Sort both strings

s = Raaghv   t = Raaghv

equate them

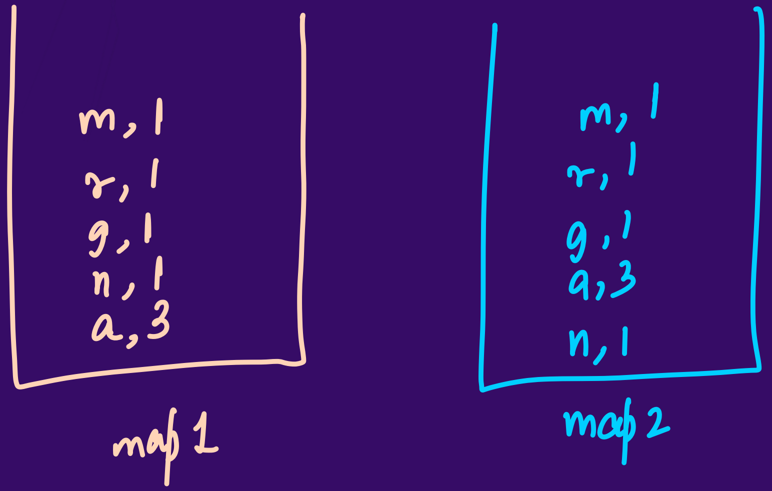
[Leetcode 242]

Ques:

map < char, freq >

Q : Valid Anagram

s = " anagram "   t = "nagaram"



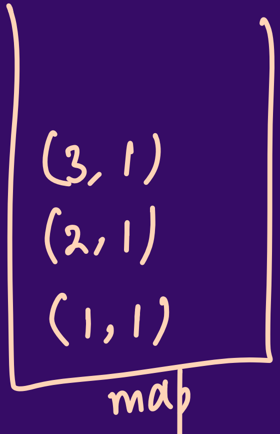
[Leetcode 242]

Ques:

Q : Valid Anagram

Frequency Map Creation

arr = { 1, 2, 3, 2, 2, 1, 4, 3, 3, 1 }



```
if(map.containsKey(arr[i])){
    int freq = map.get(arr[i]);
    map.put(arr[i], freq+1);
}
else mp.put(arr[i], 1);
```

[Leetcode 242]

Ques:

Q : Two Sum

$$arr = \{ \overset{0}{2}, \overset{1}{5}, \overset{2}{9}, \overset{3}{4} \} \qquad target = 9$$

$$ans = \{1, 3\}$$

Brute Force → T.C. = $O(n^2)$

S.C. = $O(1)$

Hashmap → T.C. = $O(n)$

S.C. = $O(n)$

[Leetcode 1]

Ques:

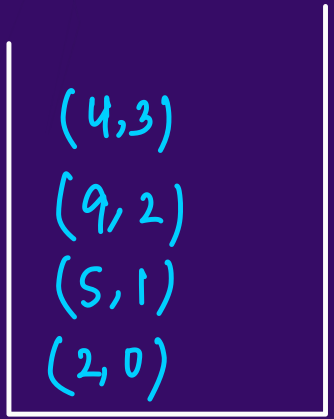
map <key, val>
ele idx

Q : Two Sum

0	1	2	3	4
2	5	9	4	1

arr = { 2, 5, 9, 4, 1 } target = 10

remaining = target - arr[i] ;



(4,3)
(9, 2)
(5, 1)
(2, 0)

map

[Leetcode 1]

TreeSet and TreeMaps

↓

insert, remove, search → O(logn)

Sorted

TreeSet / TreeMap → ordered set / ordered map

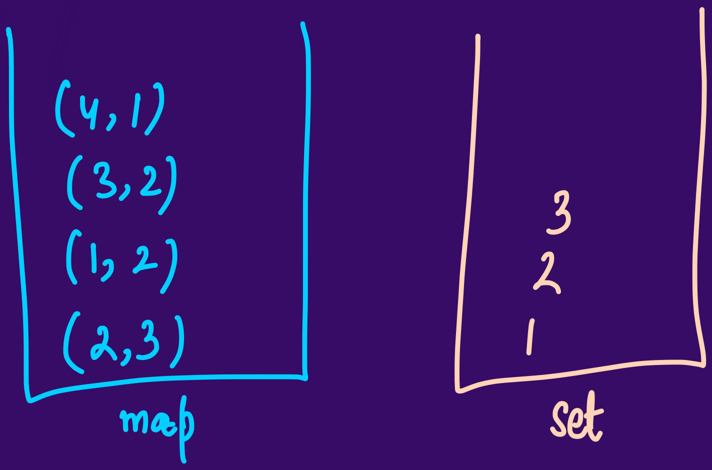
HashSet / Hashmap → unordered

Ques:

Q : Unique Number of Occurrences

arr = { 2, 2, 1, 2, 3, 3, 1, 4 }

Compare sizes of map & set



[Leetcode 1207]

Ques:

Q : Finding 3-Digit Even Numbers [Good Question]

arr = {1, 3, 2}

1 3 2

3 1 2

arr = {4, 1, 2}

1 4 2 1 2 4

4 1 2 2 1 4

arr = {1, 0, 2}

1 0 2 1 2 0

2 1 0

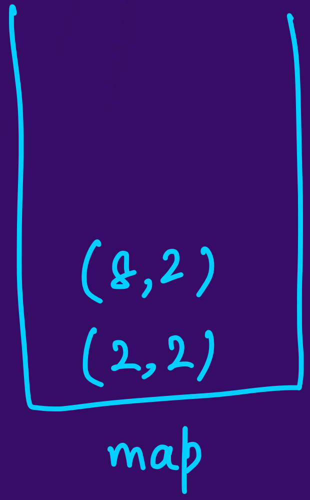
[Leetcode 2094]

Ques:

T.C. = $O(n + 500)$

Q : Finding 3-Digit Even Numbers

arr = { 2, 2, 8, 8 }



```
for(i=100 to 999){
    x = 1 4 7
        ↓ ↓ ↓
        a b c

    c = x%10 , x/=10
    b = x%10 , x/=10
    a = x
}
```

[Leetcode 2094]

Ques:

map < char, int >
↓
index

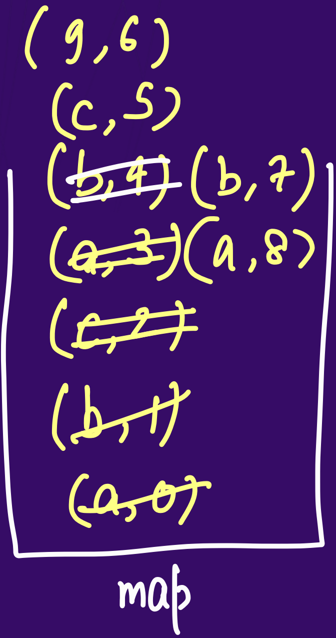
Q : Longest Substring without Repeating Characters

	0	1	2	3	4	5	6	7	8	9
s =	a	b	c	a	b	c	g	b	a	↑
						i				j

maxlen = ~~0~~ ~~3~~ 4

len = j-i

maxlen = max(len, maxlen)


(g,6)
(c,5)
~~(b,4)~~ (b,7)
~~(a,3)~~ (a,8)
~~(c,2)~~
~~(b,1)~~
~~(a,0)~~
map

[Leetcode 3]

Ques:

Q : Check if array pairs are divisible by K.

arr = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}   K = 5

→ (1,9), (2,8), (3,7), (4,6), (5,10)

→ (1,4) (2,3) (6,9) (7,8) (5,10)

2 + 3 = 5 ✓

[Leetcode 1497]

Ques:

Q : Check if array pairs are divisible by K.

a & b. and I need to figure out if $(a+b)\%k == 0$

$$\begin{tikzcd} a \arrow[d] \arrow[dd, bend right] & b \arrow[d] \\ 5p+a_1 & 5q+a_2 \\ a\%k + b\%k == 0 & \end{tikzcd}$$

[Leetcode 1497]

Ques:

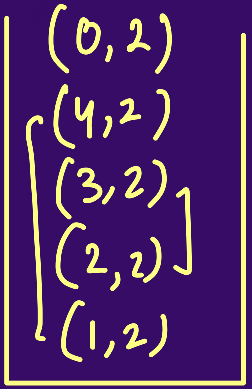
$arr[i] \% K$

Q : Check if array pairs are divisible by K.

arr = { 1, 2, 3, 4, 6, 7, 8, 9, 5, 10 } K=5

1 2 3 4 1 2 3 4 0 0

→



key = 0

rem = K - key = 5.

[Leetcode 1497]

Ques:

Q : Check if array pairs are divisible by K.

arr = { 2, 2, 3, 1, 2 }   K=4

2 2 3 1 2

(1 , 1)

(3, 1)

(2, 3)

[Leetcode 1497]

Ques:

$ele = arr[i] \% K$ $\boxed{ele += K}$

Q : Check if array pairs are divisible by K.

$arr = \{-1, 1, -2, 2, -3, 3, -4, 4\}$ $K=3$

(0,2)
(2, 1)
(-2, 1)
(1, 2)
(-1, 2)

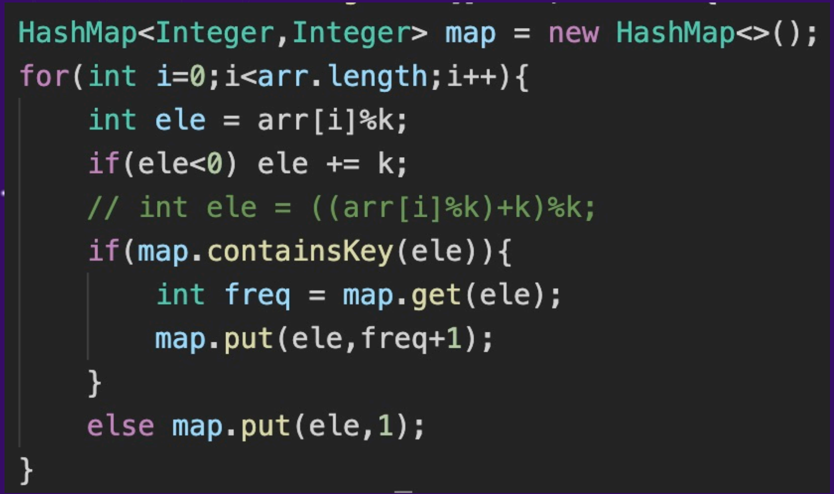
map

$$(-a) \% b = -[a \% b]$$

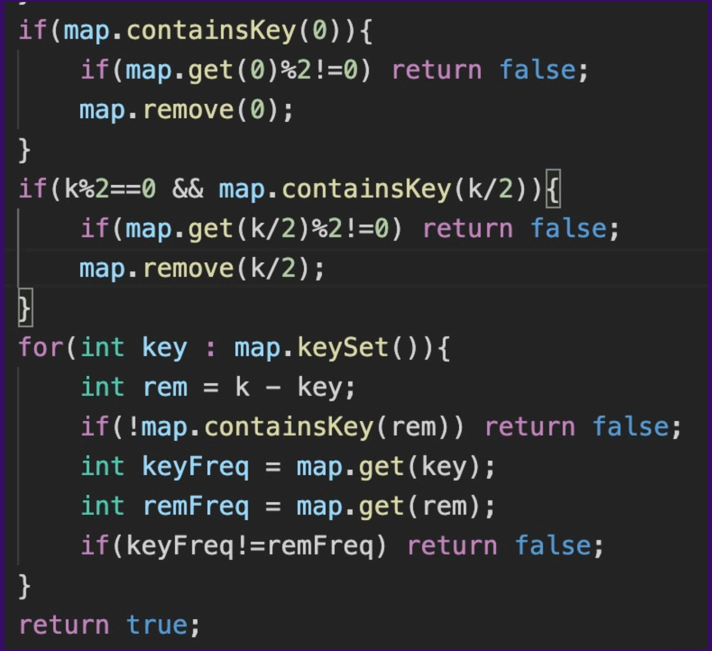
[Leetcode 1497]

Ques:

Q : Check if array pairs are divisible by K.



```java
HashMap<Integer,Integer> map = new HashMap<>();
for(int i=0;i<arr.length;i++){
    int ele = arr[i]%k;
    if(ele<0) ele += k;
    // int ele = ((arr[i]%k)+k)%k;
    if(map.containsKey(ele)){
        int freq = map.get(ele);
        map.put(ele,freq+1);
    }
    else map.put(ele,1);
}
```

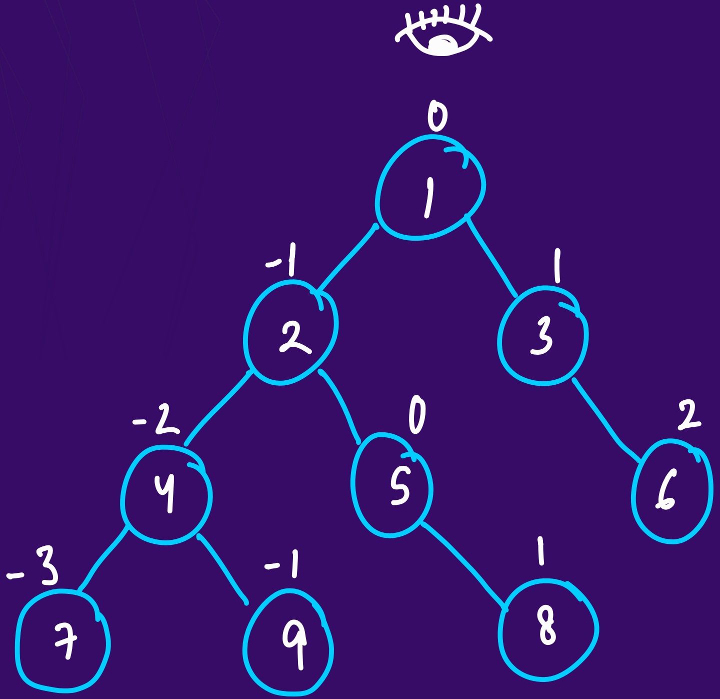


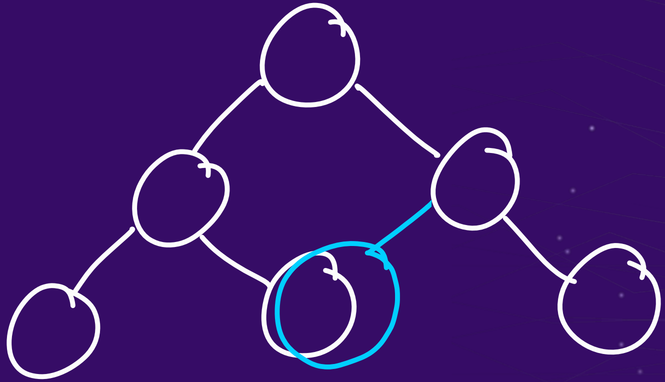
```java
if(map.containsKey(0)){
    if(map.get(0)%2!=0) return false;
    map.remove(0);
}
if(k%2==0 && map.containsKey(k/2)){
    if(map.get(k/2)%2!=0) return false;
    map.remove(k/2);
}
for(int key : map.keySet()){
    int rem = k - key;
    if(!map.containsKey(rem)) return false;
    int keyFreq = map.get(key);
    int remFreq = map.get(rem);
    if(keyFreq!=remFreq) return false;
}
return true;
```

[Leetcode 1497]

Ques:

Q : Top View of Binary Tree



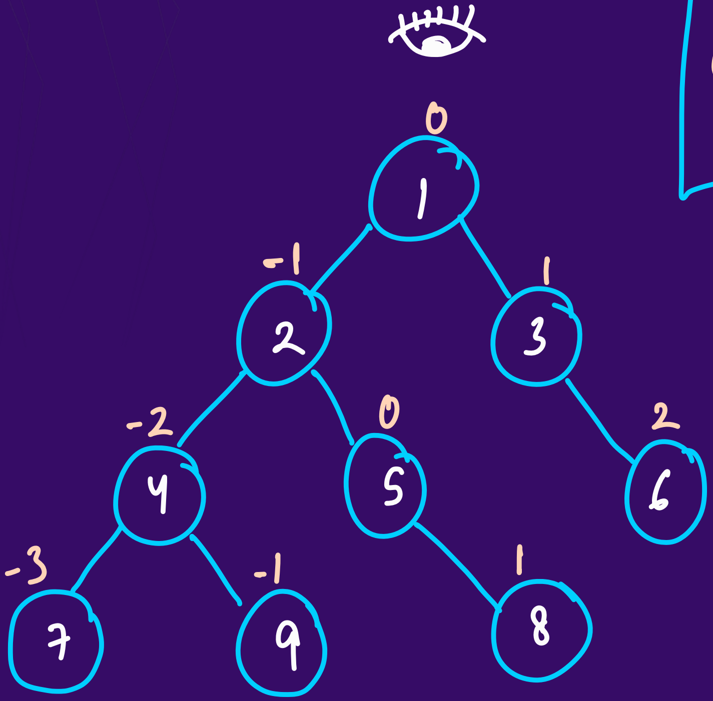


[Hackerrank]

map< level, node·val >

Ques:

Q : Top View of Binary Tree



(-3,7)
(2,6)
(-2,4)
(1,3)
(-1,2)
(0,1)

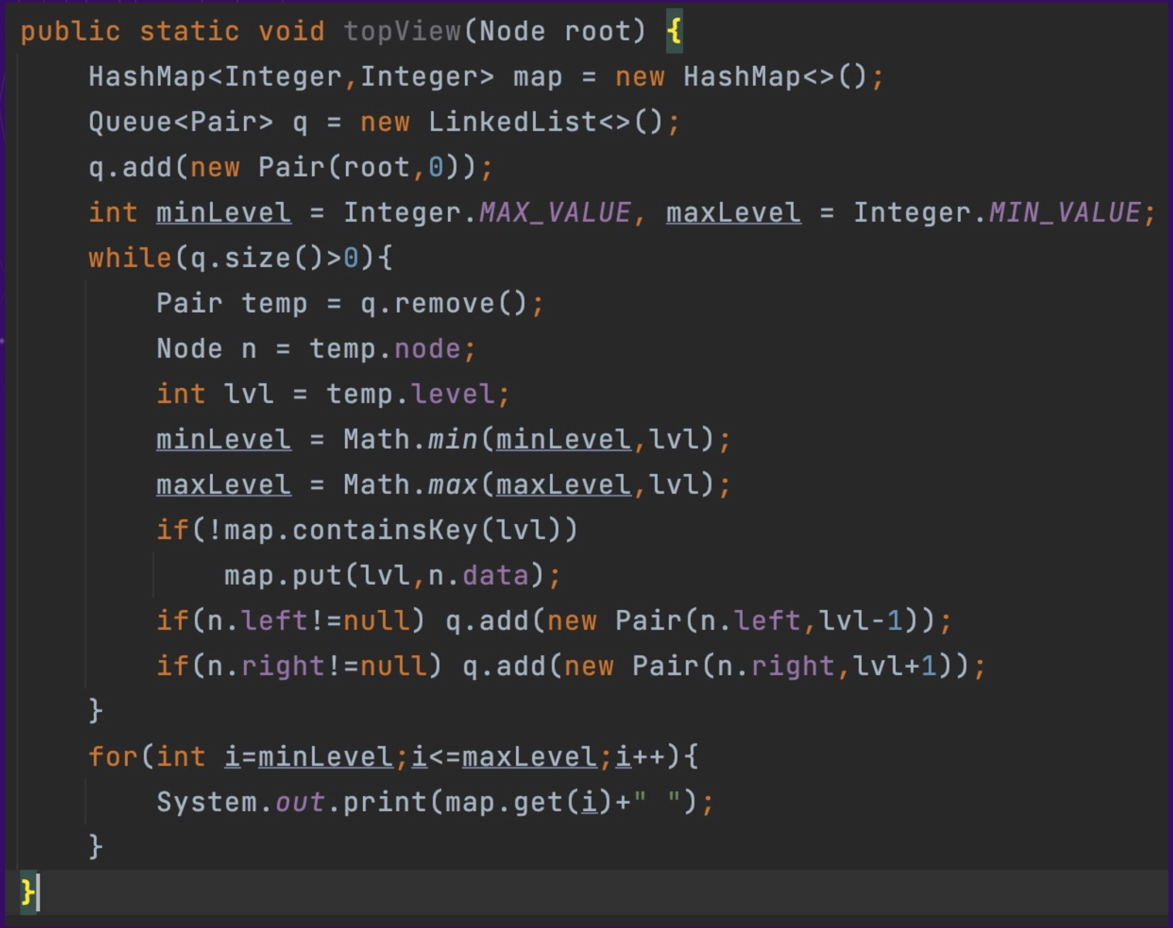
map

BFS

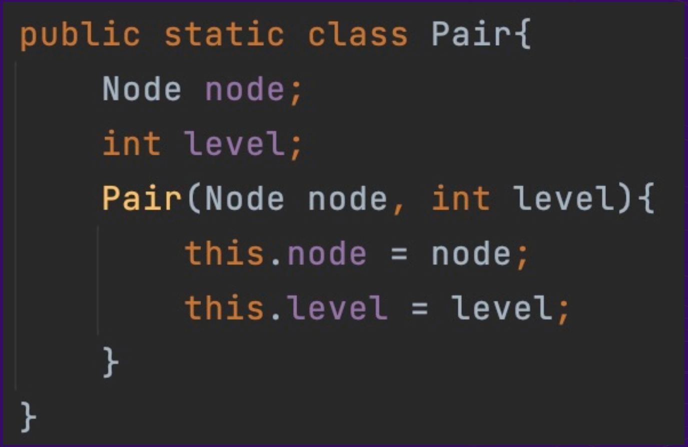
map·get (i) [i→ -3 to 2]

Queue <Node, level>

[Hackerrank]



```java
public static void topView(Node root) {
    HashMap<Integer,Integer> map = new HashMap<>();
    Queue<Pair> q = new LinkedList<>();
    q.add(new Pair(root,0));
    int minLevel = Integer.MAX_VALUE, maxLevel = Integer.MIN_VALUE;
    while(q.size()>0){
        Pair temp = q.remove();
        Node n = temp.node;
        int lvl = temp.level;
        minLevel = Math.min(minLevel,lvl);
        maxLevel = Math.max(maxLevel,lvl);
        if(!map.containsKey(lvl))
            map.put(lvl,n.data);
        if(n.left!=null) q.add(new Pair(n.left,lvl-1));
        if(n.right!=null) q.add(new Pair(n.right,lvl+1));
    }
    for(int i=minLevel;i<=maxLevel;i++){
        System.out.print(map.get(i)+" ");
    }
}
```



```java
public static class Pair{
    Node node;
    int level;
    Pair(Node node, int level){
        this.node = node;
        this.level = level;
    }
}
```

[Hackerrank]

Homework:

Q : Bottom View of Binary Tree

Ques:

Q : Count Nice Pairs in an Array

$(i, j) \quad i < j$

arr = $\{42, 11, 1, 97\}$

Brute Force → T.C. $= O(n^2)$

S.C. $= O(1)$

$$nums[i] + rev(nums[j]) == nums[j] + rev(nums[i])$$

$$nums[i] - rev(nums[i]) == nums[j] - rev(nums[j])$$

[Leetcode 1814]

Ques:

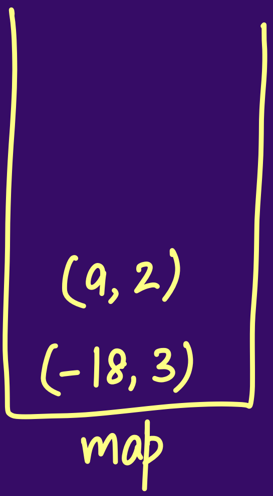
Q : Count Nice Pairs in an Array

$$nums[i] - rev(nums[i]) == nums[j] - rev(nums[j])$$

$count = \cancel{0}\ \cancel{1}\ \cancel{3}\ 4$

$$arr = \{13, 10, 35, 24, 76\}$$

$$\begin{array}{ccccc} 13 & 10 & 35 & 24 & 76 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ -18 & 9 & -18 & -18 & 9 \end{array}$$



(9, 2)

(-18, 3)

map

(13, 35) (13, 24) (35, 24) (10, 76)

[Leetcode 1814]

Ques:

Q : Count Nice Pairs in an Array

$$arr = \{ \begin{array}{ccccccc} 1, & 5, & 2, & 3, & 4, & 6, & 7 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \}$$

If arr.length == n, how many max no. of pairs can be there

$$n-1 + n-2 + \cdot \cdot \cdot \cdot \cdot\ 1 = \frac{n(n-1)}{2}$$

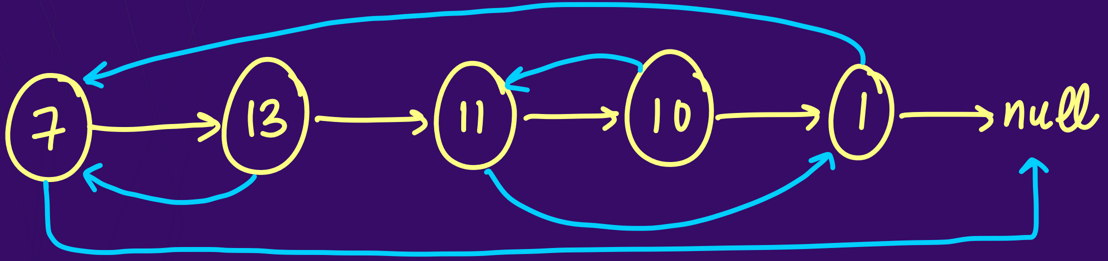
$n <= 10^5$

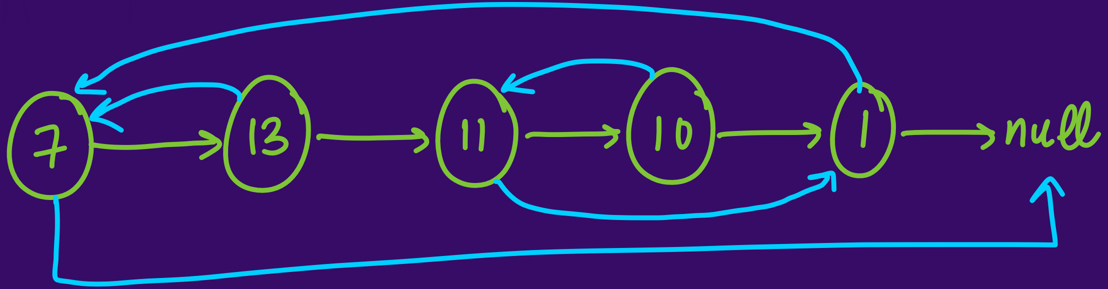
$$\boxed{\frac{10^{10}}{2}} \rightarrow$$

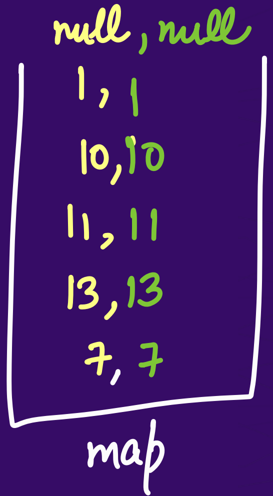
[Leetcode 1814]

Ques: HashMap < Node , Node > map ;
ori dup

Q : Copy List with Random Pointer







Extra space = O(n)
(map)

T.C. = O(n)

[Leetcode 138]

dup.random = map.get(ori.random) ;

Homework:

Q : Max Number of K-Sum Pairs

[Leetcode 1679]

Ques:

Q : Unique Length-3 Palindromic Subsequences

Str = "abcaaabcccb"

⇒ aaa bab ccc

aba bcb cac

aca bbb cbc

aabca

aaa

aba

aca

[Leetcode 1930]

Ques:

Q : Unique Length-3 Palindromic Subsequences

Hint : Try all 26 lowercase alphabets / Try all chars in string

Find out first & last occurence of each alphabet

Hashset / Frequency Array

[Leetcode 1930]

Ques:

Q : Unique Length-3 Palindromic Subsequences

str =

0	1	2	3	4	5	6
a	b	g	c	a	b	g

count

~~0~~ ~~3~~ ~~5~~ 6

a _ a

b _ b

g _ g

(c,3)
(g,2)
(b,1)
(a,0)

fmap

(a;4)
(c,3)
~~(g,2)~~ (g,6)
~~(b,1)~~ (b,5)
~~(a,0)~~

lmap

[Leetcode 1930]

Homework:

Q : Sum of Beauty of all Substrings

[Leetcode 1781]

Hashmap theory

↓

Hashing, Collision → OA

Collision → chaining

THANK YOU