unordered_map & unordered_set

Maps & Sets

Part – 2

Raghav Garg

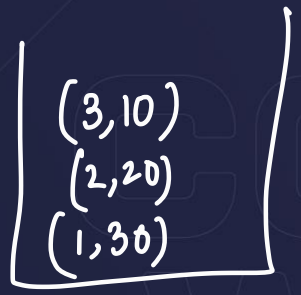
ordered vs unordered

```
S.insert(4);
S.insert(2);
S.insert(3);
```

ordered set

```
set<int> s;
map<a,b> m;
```

1 2 5 6 8



Ques: Finding 3-Digit Even Numbers [Leetcode - 2094]

3-Digit → 100 to 999

[Leetcode - 2094] → Medium

{ 2, 1, 3, 0 }

2 1 3 | 2 3 1 | 1 3 2 | 1 2 3

3 1 2 | 3 2 1

2 1 0 | 2 0 1 | 1 2 0 | 1 0 2

0 1 2 | 0 2 1

Ques: Finding 3-Digit Even Numbers [Leetcode - 2094]

v = {2, 1, 3, 0}

100 to 999

s = {2, 1, 3, 0}

i = 100

x = i = 100

a = x%10 → a = 0

x = x/10 ⇒ x = 10

b = x%10 → b = 0

x = x/10 ⇒ x = 1

c = x%10 → c = 1

i = 101

x = i = 101

a = 1

b = 0

c = 1

i = 102

x = 102

a = 2

b = 0

c = 1

Ques: Finding 3-Digit Even Numbers [Leetcode - 2094]

{2, 2, 8, 8, 2}

→ {222, 228, 282, 822
882, 828, 288}

Ques: Finding 3-Digit Even Numbers [Leetcode - 2094]

arr = {2, 2, 8, 8, 2}

100 to 999

m

(2,0)
(2,1)
(2,2)
(8,2)
(2,3)

i = 222
x = 222

a = 2
b = 2
c = 2

Ques: Count Nice Pairs in an Array [Leetcode - 1814]

0 1 2 3

{42, 11, 1, 97}

(0,3) → 42 + 79 == 97 + 24

= 121 = 121

(1,2) = 11 + 1 = 12

11 + 1 = 12

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

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

Ques: Count Nice Pairs in an Array [Leetcode - 1814]

{42, 11, 1, 97}

⇒ {18, 0, 0, 18}

for every 'i' in nums →

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

nums = {13, 10, 35, 24, 76}

↳ {-18, 9, -18, -18, 9}

(9, 2)
(-18, 3)
(-18, 2)
(9, 1)
(-18, 1)

Count = 6 ✗ 3 (4)

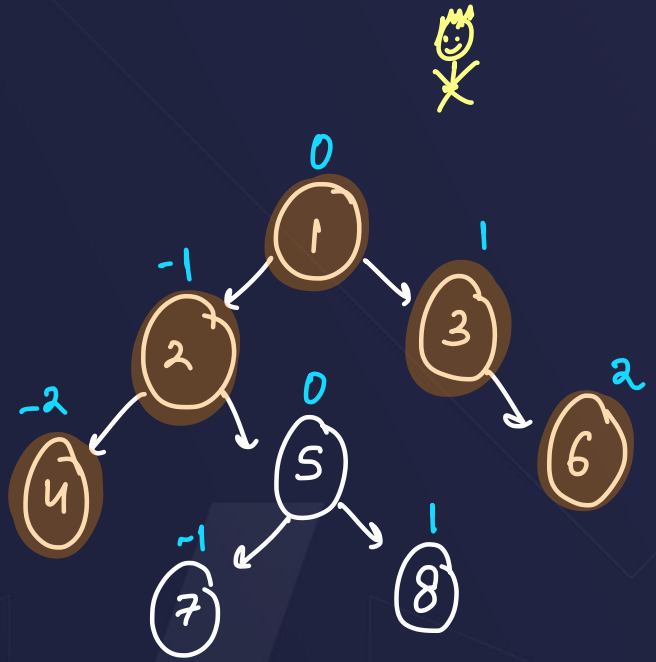
13 - 31 = -18

35 - 53 = +18

24 - 42 = -18

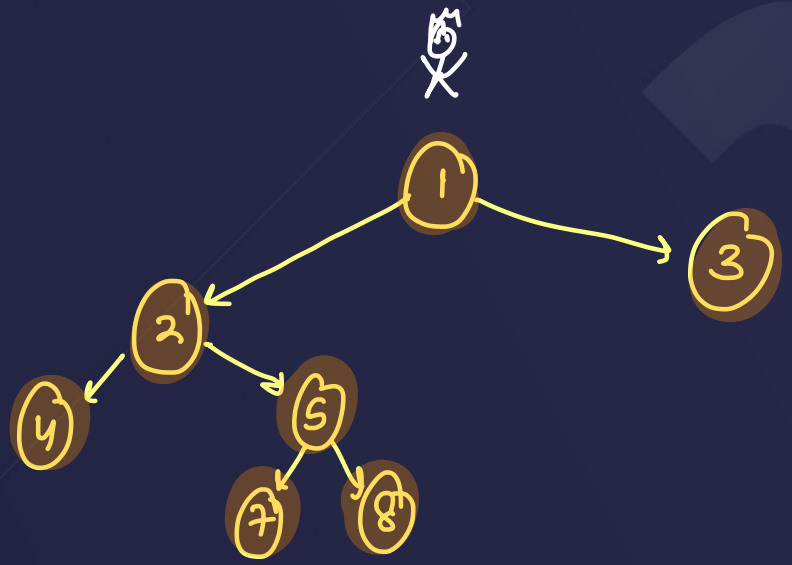
76 - 67

Top view of binary tree



4 2 1 3 6

✓



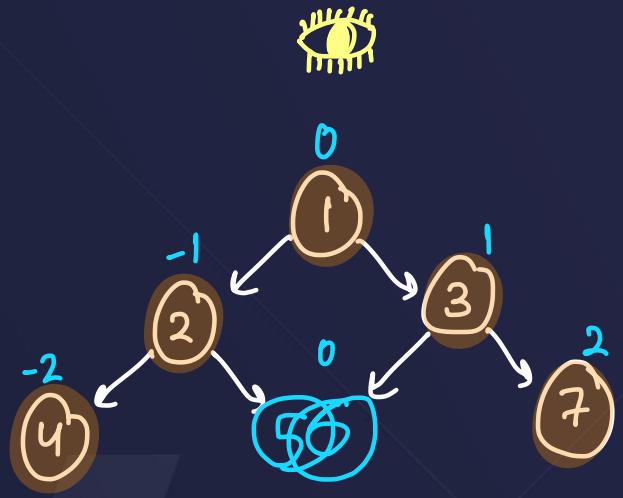
4 2 7 5 8 1 3

✗

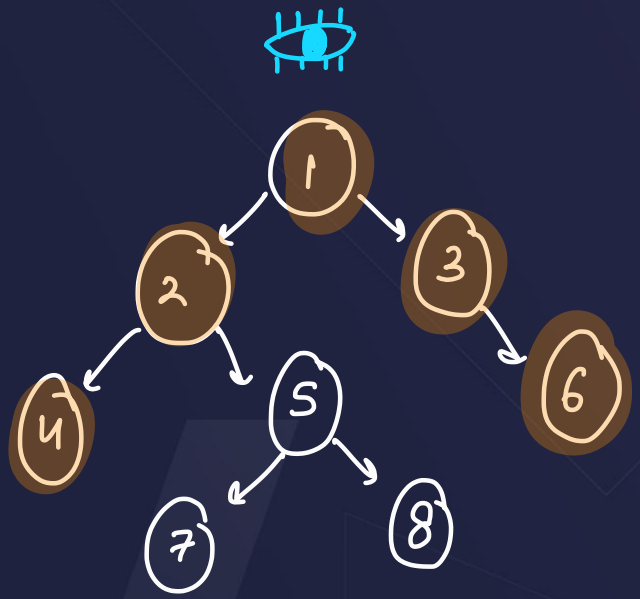
Top view of binary tree

Hints : 1) Hashmap

2) Level Order Traversal



Top view of binary tree



map < int , int > m;
level value

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

m

<3,1>

1

,<5, 0>

Ques: Copy List with Random Pointer [Leetcode - 138]

1) Create deep copy without random connections
2) Alternate linking
3) Mark random pointers
4) Remove alternate connections

$S.C. = O(1)$

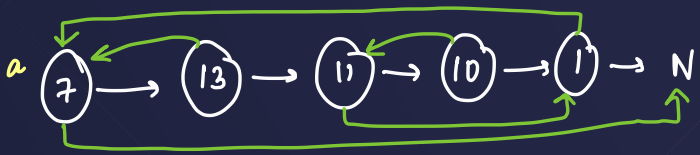
$T.C. = O(n)$

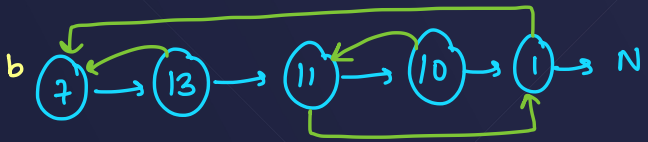
Method-2 :

↓

$T.C. = O(n)$

$S.C. = O(n)$





map < Node*, Node* >

a → key nodes

b → key nodes

a	b
(1, 1)	
(10, 10)	
(11, 11)	
(13, 13)	
(7, 7)	

m

THANK YOU!