

2 Sum : unique Pair

Saturday, 18 September 2021

2:23 PM

target = 10

target sum pair \rightarrow distinct elements in array. \rightarrow Ins, BST.

array \rightarrow 1 2 2 6 3 1 5 5 8 8 7 7 4 6 9 9

Sort \rightarrow 1 1 2 2 3 4 5 5 6 6 7 7 8 8 9 9

Sorted

right left

Removal of duplicate will not work

NOTE: If we can manage repetition of single variable then we can prevent repetition of pair.

check of previous similarity.

```
while (left < right) {
    sum = arr[left] + arr[right];
    if (sum == target) {
        add (arr[left] & arr[right]) in list;
        left++; right--;
    } else if (sum > target) {
        right--;
    } else {
        left++;
    }
}
```

{ {1, 9}, {2, 8}, {3, 7}, {4, 6}, {5, 5} }

res → [[1,7], [2,6], [3,5]]

① ① ① 1 2 2 3 5 6 6 7 7 7 7

↑
right
↑
left

sum = ~~8~~ ~~8~~ ~~8~~ ~~8~~ 8

[2,6], [3,5]

Final Result → [[1,7], [2,6], [3,5]]

target = 8

```
Arrays.sort(arr);
int left = 0;
int right = arr.length - 1;
List<List<Integer>> res = new ArrayList<>();

while(left < right) {
    // how to prevent repetition
    if(left != 0 && arr[left] == arr[left - 1]) {
        left++;
        continue;
    }

    int sum = arr[left] + arr[right];
    if(sum == target) {
        List<Integer> subres = new ArrayList<>();
        subres.add(arr[left]);
        subres.add(arr[right]);
        res.add(subres);
        left++;
        right--;
    } else if(sum > target) {
        right--;
    } else {
        left++;
    }
}
```


3 Sum : Unique Triplets

Saturday, 18 September 2021 2:23 PM

target = 0 unique triplet which have sum is equal to target.

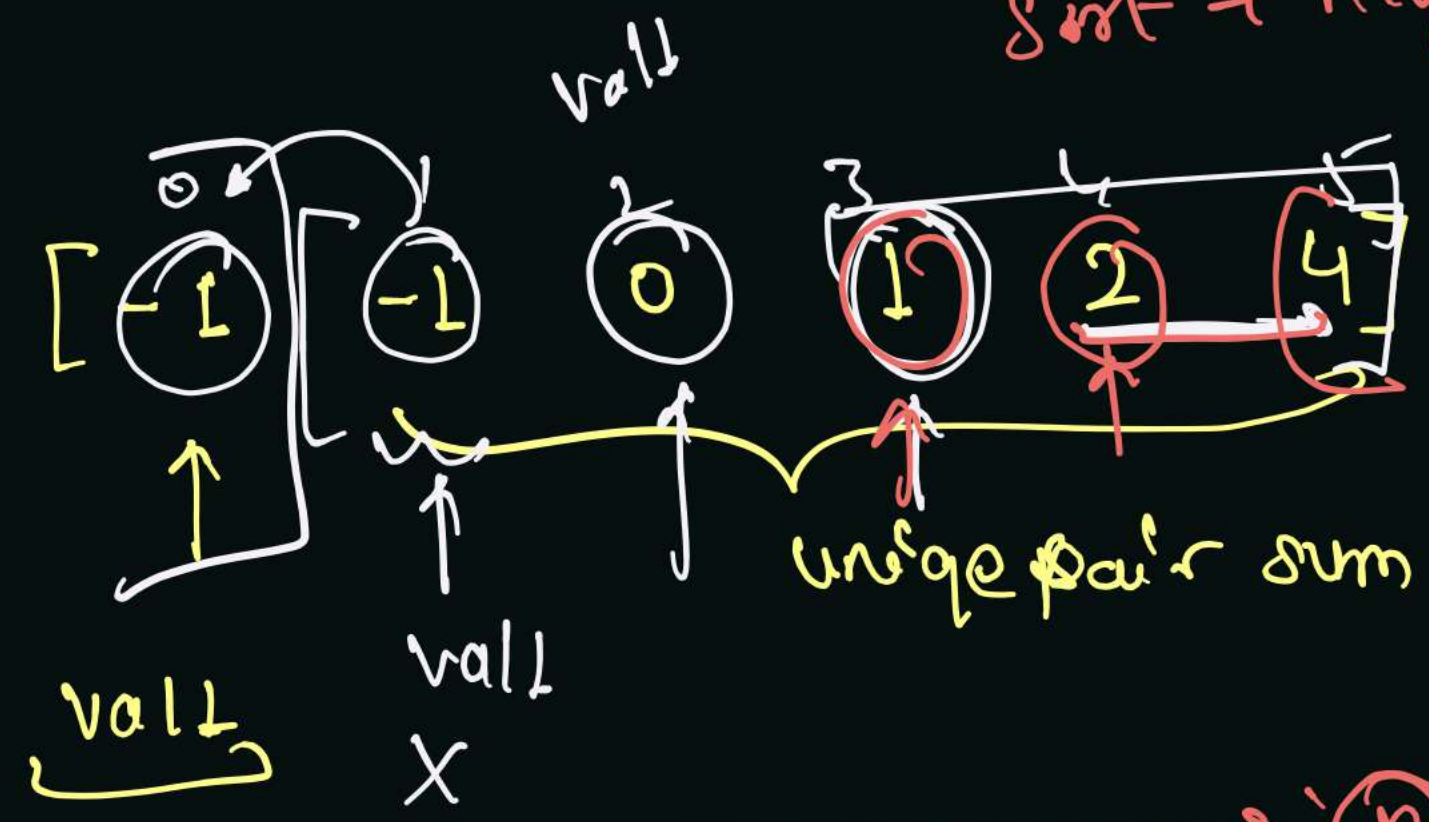
0	1	2	3	4	5
-1	0	1	2	-1	4

target

sort $\rightarrow n \log n$ 6-10 6-3

triplet = val, val2, val3

sort \rightarrow



pair with sum = 1

0	1
-1	2

val

Time

complexity

remaining target = target - val

$= 0 - (-1) = 1$

for every iterate we are traversing rest array in 'two sum' $\rightarrow O(n)$

0	1	-1
-1	2	-1

triplet

$n \times O(n) = O(n^2)$

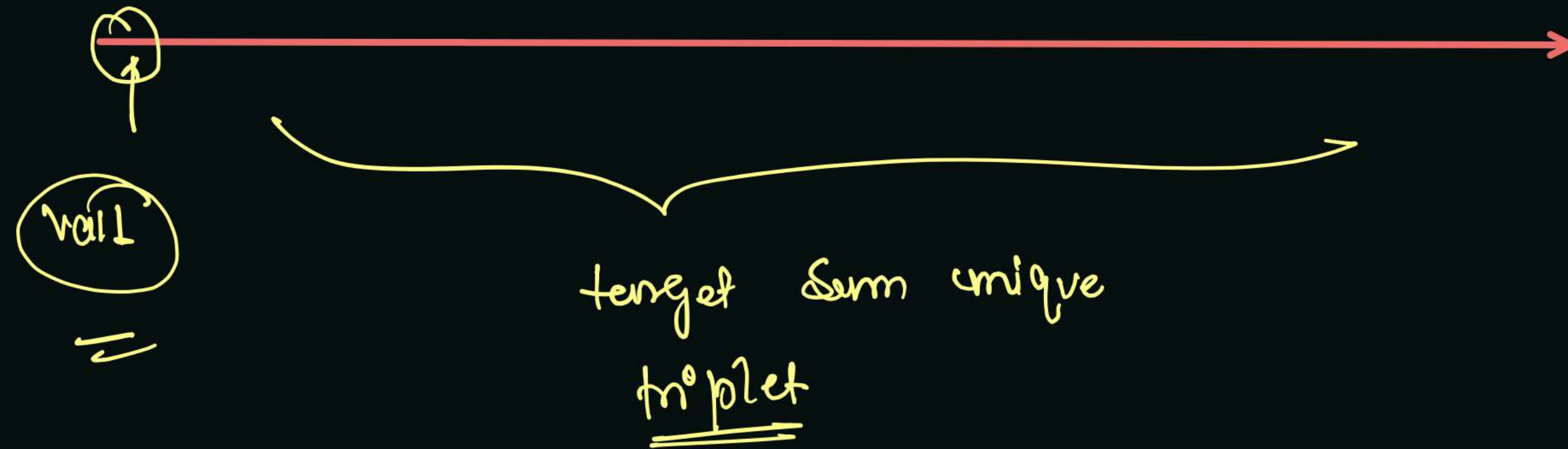
$O(n \log + n^2)$

$= O(n^2)$

4 Sum : Unique Quadruples

Saturday, 18 September 2021

2:23 PM



k sum \rightarrow same as 4sum and 3sum

\rightarrow Base case \Rightarrow k=2 then call to 2sum

Complex Number Multiplication

Sunday, 19 September 2021

2:51 PM

Complex Number \rightarrow "real + Imaginary i"

$$\underline{\underline{"0 + 2i"}} \rightarrow \text{real} = 0 \quad \text{Imaginary} = 2$$

$$\text{"-1 + 3i"} \rightarrow \text{real} = -1 \quad \text{Imaginary} = 3$$

$$\underline{\underline{"1 + -2i"}} \rightarrow \text{real} = 1 \quad \text{Imaginary} = -2$$

$$\text{"7 + 0i"} \rightarrow \text{real} = 7 \quad \text{Imaginary} = 0$$

$$a - bi$$

$$\underline{a} + \underline{-bi}$$

$$= \underbrace{(a_1 + b_1 i)}_{\text{C.N.1}} * \underbrace{(a_2 + b_2 i)}_{\text{C.N.2.}}$$

$$\underline{\underline{\text{Result}}}$$
$$\underline{\underline{a + bi}}$$

$$= a_1 \cdot a_2 + a_1 \cdot b_2 i + a_2 \cdot b_1 i + b_1 \cdot b_2 i^2$$
$$= \underbrace{(a_1 \cdot a_2 - b_1 \cdot b_2)}_{\text{Real}} + \underbrace{(a_1 \cdot b_2 + a_2 \cdot b_1)}_{\text{Imaginary}} i$$

$$\underline{\underline{i^2 = -1}} \quad \begin{matrix} a_1, b_1 \\ a_2, b_2 \end{matrix}$$

$$a = a_1 \cdot a_2 - b_1 \cdot b_2$$

$$b = a_1 \cdot b_2 + a_2 \cdot b_1$$

$$\text{result} = \underline{\underline{"a + bi"}}$$

$$\text{num1} = a_1 + b_1 i = \underbrace{7}_{1^x} + \underbrace{2i}_{1^x}$$

$$\underline{a_1} = \text{Integer.parseInt}(\text{num1.substring}(0, \text{num1.indexOf('+')}));$$

$$\underline{b_1} = \text{Integer.parseInt}(\text{num1.substring}(\text{num1.indexOf('+') + 1, \text{num1.length() - 1}));$$

Similar a_2 and b_2 .

$$\text{result} = a + bi = \underline{\underline{(a_1 a_2 - b_1 b_2) + (a_1 b_2 + a_2 b_1)i}}$$

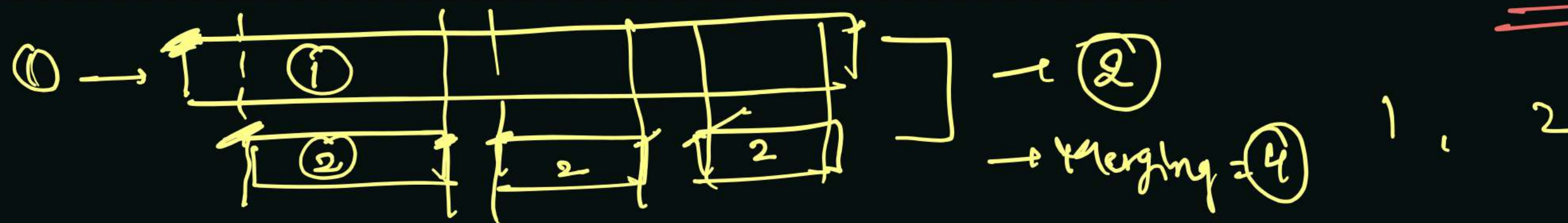
Min. Platform

Sunday, 19 September 2021

2:51 PM

⇒ Max. number of train
at a particular
time.

arr[] = {0900, 0940, 0950, 1100, 1500, 1800}
dep[] = {0910, 1200, 1120, 1130, 1900, 2000}



platform = 1

platform = 2

platform = 0 1 0 1 2 2 1 0 1 2 1 0

max = 3 → min No. of platform = 3



Total platform = 3 Min

arr → 09:00 09:40 09:50 11:00 15:00 18:00

dep → 09:10 12:00 11:20 11:30 19:00 20:00

Sort arrival and departure → (for sorted time frame)

<u>arrival</u>	<u>departure</u>
→ 09:00	09:10
09:40	11:20
09:50	11:30
11:00	12:00
15:00	<u>19:00</u>
18:00	20:00

j →

i →

platform = ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~2~~ ~~2~~ ~~2~~ ~~1~~ ~~0~~ ~~1~~ 2

max = ~~0~~ ~~1~~ ~~2~~ (2) ^{max}

Result = 3

```
while(i < arr.length) {  
    if(arr[i] <= dep[j]) {  
        platform++;  
        i++;  
    } else {  
        platform--;  
        j++;  
    }  
    max = Math.max(max, platform);  
}
```