

Working in Programming Using Sets & maps

2 Sum

Suppose:

$f = [5, 2, 11, 7, 15]$ $tar = 9$

$$7 + 2 = 9$$

There pairs can be
more than one as well

Approaches

- ① Brute force
- ② Better
- ③ Optimized

① Brute force

① All pairs extraction

```
for(i=0; i<n; i++) {  
    first = arr[i]  
    for(j=i+1; j<n; j++) {  
        second = arr[j]  
        sum = arr[i] + arr[j]  
        if (sum == tar)  
            ANS {}  
    }  
}
```

T.C
 $O(n^2)$

② Better

$[2, 5, 7, 11, 15]$
start $\xrightarrow{9}$ end
sum = start + end
end -

Steps

- ① Sort array
- ② while (start < end) {
 sum = arr[start] + arr[end]
 if (sum == target)
 return indexes;
 else if (sum < target)
 start++;
 else
 end--;
}
- ③ sum = target
- ④ Return indexes.

Optimized

arr = [5, 2, 11, 7, 15] tar = 9

pairsum = first + second = tar

→ second = target - first

for (i=0; i < n; i++)

first = arr[i]

↓
1st step

2nd step

Store them in unordered map

<value, idx>

5	0
2	1
11	2

arr [5, 2, 11, 7, 15]

first

second = tar - first

① Take the first one

② Find the comp (2nd one)

③ Store the first in unordered map

9 - 5 = 4 ✗

9 - 2 = 7 ✗

9 - 11 = -2 ✗

9 - 7 = 2 ✓

second = 2

Found!

Index → [3, 1]

Pseudocode

unordered map <i, i> map

for (i=0; i < n; i++) {

first = arr[i]

sec = tar - first

if (map.find(sec) != map.end())

return {i, map(sec)}

map[first] = i

Find missing & Repeating values

grid = $[[9, 1, 7], [8, 9, 2], [3, 4, 6]]$

9	1	7
8	9	2
3	4	6

$n \times n$

$[1, n^2]$

a 2(times)

$[1, 9]$

b (missing)

$[9, 5]$ 5

Pseudocode

vector<int> sum

$n = \text{grid.size}()$

a, b

for(int i=0; i<n; i++) {

for(j=0; j<n; j++) {

actualsum += grid[i][j]

ans.pushback

s.insert(grid[i][j])

expSum = $(n \times n) \times (n \times n) / 2$

$b = \text{expSum} + a - \text{actualSum}$

Find Duplicate (Two pointers)

do { slow = arr[slow] + 1

fast = arr[arr[fast] + 1]

while (slow != fast) {

slow = (slow != fast) {

slow = arr[slow] + 1

slow = arr[0]

while (slow != fast) {

slow = arr[slow] + 1

fast = arr[fast] + 1