

Two Pointers

Three Sum Problem

17th August, 2022

Given an array and a value, find if there exists three numbers whose sum is equal to the given value.

eg Input: $n=6$, target = 24

array elements: 12 3 7 1 6 9

output: True, (12, 3, 9 are the 3 elements whose sum is equal to 24)

eg Input: $n=6$, target = 5

array elements: 12, 3, 7, 1, 6, 9

output: false, (No elements are present)

Brute Force

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

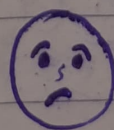
```
    for (int j = i + 1; j < n; j++)
```

```
        for (int k = j + 1; k < n; k++)
```

```
            if (arr[i] + arr[j] + arr[k] == t)
                found = true;
```

Time complexity: $O(N^3)$

Can we do better?



??

Two Pointer Technique

- * Sort the array
- * Traverse the array and fix the first element of the triplet. The problem reduces to find if there exist two elements having sum equal to $x - \text{array}[i]$.

eg target = 24 , element to find = $24 - 1$

1	3	6	7	9	12
---	---	---	---	---	----

↑

eg target = 24 , element to find = $24 - 3$

1	3	6	7	9	12
---	---	---	---	---	----

↑

Time complexity:

sorting the array: $O(N \log N)$

C standard की array sort करने के लिए qsort function की जरूरत नहीं पड़ती whereas C++ एक hybrid algorithm use करता है sort करने के लिए which is hybrid algorithm. (introsort = quicksort + heap sort).

New C++ 11 uses sorting complexity to be $O(N \log N)$.
Previous C++ 03 uses $O(N^2)$ as worst case and average case to be $O(N \log N)$.

Iterating each element and applying two sum
problem to the rest of the array : $O(N^2)$

Final Time Complexity : $O(N^2)$