# Triplet Sum in an array

Problem: Find all unique triplets (a,b,c) in an array such that a+b+c = target sum

e.g arr = [1, 4, 6, 2, 3, 8, 5]

| 1 | 4 | 6 | 2 | 3 | 8 | 5 |
|---|---|---|---|---|---|---|

target = 10.

**Step1:** Sorting the array

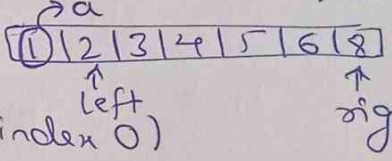
Use sort method in C++ or to sort an array

Sorted arr = [1, 2, 3, 4, 5, 6, 8]

| 1 | 2 | 3 | 4 | 5 | 6 | 8 |
|---|---|---|---|---|---|---|

**Step2:** Fix 1 element & use 2 pointers

we fix one element (a) & find 2 element (b,c) using 2 pointers.




**Iteration 1:** Fix ~~A~~ a=1 (at index 0)

Target sum needed = 9 (10-1)

→ Use 2 pointers:

left → pointing to index 1 (element 1)

right → pointing to index 6 (element 8)

→ Check sum

2+8 = 10 (Too large) → move right --

2+6 = 8 (Too small) → move left ++

3+6 = 9 (Matched → Add [1, 3, 6])

move both pointers (left++, right--) to find another set of 3 elements. (if exist)

→ now pointers are pointing

left → pointing to 4 ✓ right → pointing to 5

- Check sum

  4+5=9 (matched → Add [1,4,5])

  - move both pointer (left++, right--)
  - But now ~~err~~ left > right → so stop

similarly iterate for A=2 (fix element at index 1)

Similary iterate for A=3 (fix element at index 2)

" iterate for A=4 (fix element at index 3)

→ Sorting of array takes O(nlogn)

→ 2 pointer technique will take $O(n^2)$

→ If you print those triplets without storing them in 2D array, it will require O(1) space.

→ But if you store it in 2D array it will take O(k) where k → no. of triplets in an array.

In worst case $O(k) \approx O(n^2)$, if there are too many valid triplets