

Two Sum :-

2	6	5	8	11	7
0	1	2	3	4	5

target = 14

There can be two varieties :-

- 1) If there exist x and y where $x+y=14$ where x and y are available at different indices. So, return true or false.
- 2) Assume, there always exist such x & y where $x+y=14$. ~~Find~~ So, return the indices of x and y .

• Brute Force :-

Iterate over the array. And for each element i check if there exist j where $arr[i] + arr[j] = \text{target}$.

$$T = O(n^2); S = O(1)$$

1st Best Approach :-

i	i	i	i		
2	6	5	8	11	7
0	1	2	3	4	5

Hash Map

(2, 0)
(6, 1)
(5, 2)

target = 14

$$\begin{aligned} & \left[\begin{aligned} 2 + x &= 14 \\ \Rightarrow x &= 14 - 2 = 12 \\ \Rightarrow x &= 12 \text{ (Search for 12 in Map)} \end{aligned} \right. \\ & \quad \text{if not found then} \\ & \quad \text{put } x \text{ in the map} \\ & \left[\begin{aligned} x &= 14 - 6 \\ &= 8 \end{aligned} \right. \end{aligned}$$

$$\left[\begin{aligned} x &= 14 - 5 \\ &= 9 \end{aligned} \right.$$

Here, hasMap is used to keep track of the elements which are already visited.

$$\left[\begin{aligned} x &= 14 - 8 \\ &= 6 \end{aligned} \right] \Rightarrow 6 \text{ is found in the Map}$$

$$T = O(n) * O(1) = O(n)$$

$$S = O(n)$$

2nd Best Approach (Two pointer)

⇒ Sort the array

i	j	i		j	j
2	5	6	7	8	11
0	1	2	3	4	5

target = 14

$$2 + 11 = 13 < 14 \quad (i++)$$

$$5 + 11 = 16 > 14 \quad (j--)$$

$$5 + 8 = 13 < 14 \quad (i++)$$

$$6 + 8 = 14 == 14$$

$$T = O(n \log n) + O(n) = O(n \log n)$$

$$S = O(1)$$

In this approach we are changing the array. So, we can say true or false but we can't find the indices.

Note :-

There can be a third variety of this problem. We have to find all unique pairs and also in a pair same element can't be taken twice.

We can use 1st and 2nd best approach (Both) to find all unique pairs.

↳ We just have to make slight modification

• In 1st Best Approach, we will take a Hashset

Here, we won't stop the loop as soon as we get a valid pair. We will continue the loop and keep on adding elements in hashMap.

During the looping, ~~we~~ as soon as we find a valid pair, we will ~~keep the pair~~ ~~in hash~~ sort the elements of the pair and store it in hashset.

HashSet will itself make sure that the pairs are unique. Because it stores only unique elements.

• In 2nd Best Approach, we will use the same

logic as previous. Additionally, we have to do $i++$ and $j--$ when we find a pair and keep on adding the pairs to a list.

There is an edge case here, when some of the elements are repeated:-

2	4	5	5	5	5	8	9	11
		i			j			

In such case, we compare $(arr[i-1] == arr[i])$
 $\& \&$

$$(arr[i] = arr[j+1])$$

we keep on doing $i++ \& j--$

\Rightarrow Check codes for better understanding.