array = [1,3,5,6,7] target = 8

Time nlogn | Space n

⇒ 1. sort ⇒ [1,3,5,6,7] nlogn, n
⇒ 2. sort ⇒ [7,4,3,2,1] → reverse.
⇒ 3. Check for same number using two pointers O[n]

⇒ Check if two sorted arrays have the same number.

→ a = [1,3,5,6,7]
b = [2,4,4,5,9,10]

~~i = 0~~
~~j = 0~~
i, j = 0, 0

~~if a[i] < b[j]~~

while (i < len(a) and j < len(b)):
    if a[i] < b[j]:
        i += 1
    if a[i] > b[j]:
        j += 1
    if a[i] == b[j]:
        return True
return False.

→ Time ⇒ n log(n)
Space ⇒ n

Second try:-

Time → O(n)?!

target = 8 array = [1, 3, 5, 9]

hash = 1 → [0]
3 → [1]
5 → [2]
9 → [3]

t-arr2 = [7, 5, 3, -1]

Pseudo code

→ array = a
→ t-a = s_a
→ put a in a hashmap with indices
→ add (t-a) to hashmap with indices
→ if any value in hashmap has 2 elements and the ~~number of~~ elements are not same then return those indices.