Target Sum

↳ array is not sorted
↳ may contain duplicates
↳ only print unique pairs

1) arr [3 3 5 5]

target = 8

O/P 3 5

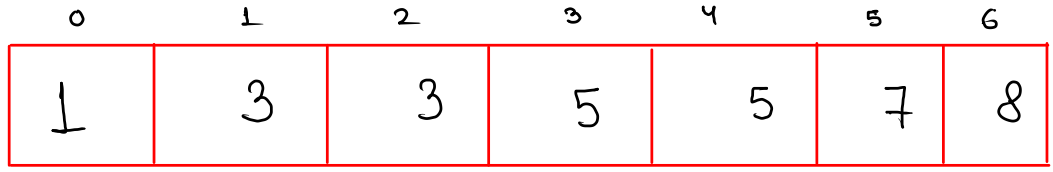
2) arr [3 3 2 4]

target = 6

O/P 2 4
3 3

Ex :-

arr



target = 8

sum = ~~9~~ ~~8~~ 8

op

1	7
3	5

psudo code

1) sort the array
2) make 2 pointers $i=0$ and $j=n-1$
3) loop until $i < j$

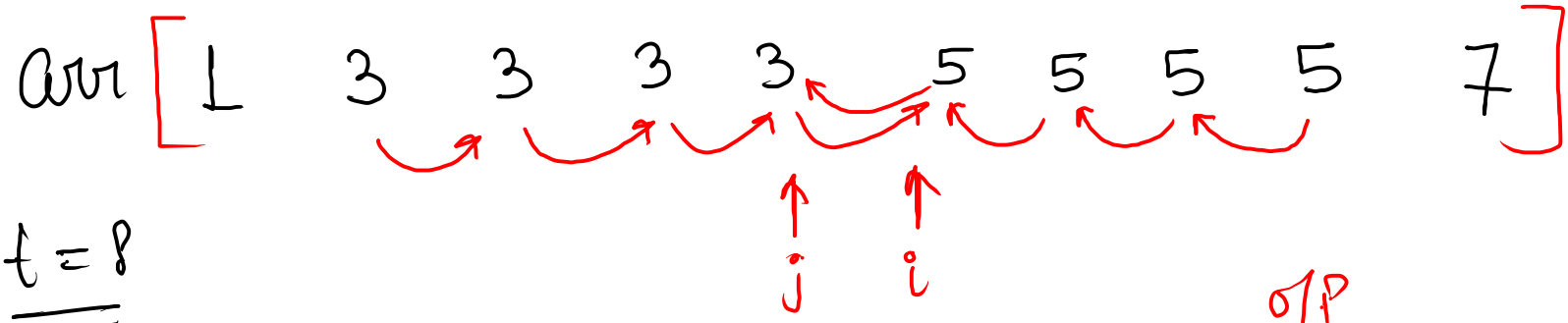
3.1) sum = arr[i] + arr[j]

3.2) sum < target
i++

3.3 sum > target
j--

3.4) sum == target

```
Syso( arr[i] + " " + arr[j] );
while ( arr[i] == arr[i+1] ){
        i++;
}
while ( arr[j] == arr[j-1] ){
        j--;
}
i++, j--;
```



t = 8

o/p

1	7
3	5

psudo code

1) sort the array
2) make 2 pointers i = 0 and j = n-1
3) loop until i < j
   - 3.1) sum = arr[i] + arr[j]
   - 3.2) sum < target
     i++
   - 3.3 sum > target
     j--
   - ✓ 3.4) sum == target

```
Syso( arr[i] + " " + arr[j] );
while ( arr[i] == arr[i+1] ){
        i++;
}
while ( arr[j] == arr[j-1] ){
        j--;
}
```

Imp → i++ ; j-- ;

Code

```java
public static void targetSum(int[] arr, int n, int target) {
    Arrays.sort(arr);
    int i = 0;
    int j = n - 1;
    while ( i < j ) {
        int sum = arr[i] + arr[j];
        if ( sum == target ) {
            System.out.println( arr[i] + " " + arr[j] );
            while ( i < j && arr[i] == arr[i + 1] ) {
                i++;
            }
            while ( i < j && arr[j] == arr[j - 1] ) {
                j--;
            }
            i++;
            j--;
        } else if ( sum < target ) {
            i++;
        } else {
            j--;
        }
    }
}
```

gmp

T.C = O(n log(n) + n)

≃ O(n log(n))

S.C = O(1)

3 Sum

arr =

-2	0	2	4	-2	-8
0	1	2	3	4	5

arr[i] + arr[j] + arr[K] == 0

arr[i] + arr[j] == -1 * arr[K]

target

logic

Note:- we are performing the previous task multiple times.

arr =

-8	-2	-2	0	2	4
0	1	2	3	4	5

k → i = k+1
↳ j = n-1

k i j

k = 0 , target = 8 → sum = ~~2~~ ~~2~~ ~~4~~ 6

k = 1, target = 2 → sum = ~~2~~ 2

k = 2, target = 2 → sum = ~~4~~ 2

k = 3, target = 0 → sum = 6

-2	-2	4
-2	0	2
-2	0	2

psudo code

1) sort
2) make K pointers
3) loop for K value

3.1) same as previous question

Code

```java
public static void targetTriplet(int[] arr, int n) {
    Arrays.sort(arr);
    for (int k = 0; k < n; k++) {
        int i = k + 1;
        int j = n - 1;
        int target = -1 * arr[k];
        while ( i < j ) {
            int sum = arr[i] + arr[j];
            if ( sum == target ) {
                System.out.println(arr[k] + " " + arr[i] + " " + arr[j]);
                while ( i < j && arr[i] == arr[i + 1] ) {
                    i++;
                }
                while ( i < j && arr[j] == arr[j - 1] ) {
                    j--;
                }
                i++;
                j--;
            } else if ( sum < target ) {
                i++;
            } else {
                j--;
            }
        }

        while ( k + 1 < n && arr[k] == arr[k + 1] ) {
            k++;
        }
    }
}
```

$T.C = O(N^2 + N \log N)$

$\simeq O(N^2)$

$S.C = O(1)$

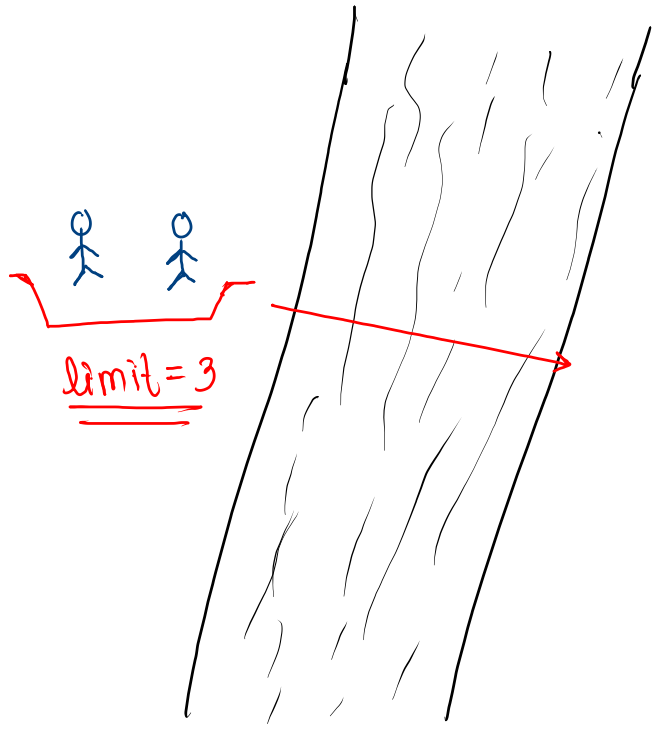
Count boat (target sum)

$n = 4$

0	1	2	3
3	2	2	1

arr = [3 2 2 1]

limit = 3



arr[i] + arr[j] <= limit

$n = 4$

	0	1	2	3
arr =	1	2	2	3
	j	i		

limit = 3

sum = ~~4~~ ~~3~~ 4

boat = ~~0~~ ~~1~~ ~~2~~ 3

psudo code

1) sort
2) i = 0 , j = n-1
3) loop until i <= j

3.1) if sum > limit
j-- , boat++j

3.2 else if sum <= limit
i++ , j-- , boat++j

Code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    int limit = scn.nextInt();
    countBoats(arr, n, limit);
}
public static void countBoats(int[] arr, int n, int limit) {
    Arrays.sort(arr);
    int i = 0;
    int j = n - 1;
    int count = 0;
    while ( i <= j ) {
        int sum = arr[i] + arr[j];
        if ( sum > limit ) {
            j--;
        } else {
            i++;
            j--;
        }
        count++;
    }
    System.out.println(count);
}
```

limit = 9

	0	1	2	3	4	5
arr	5	5	8	6	2	4

arr	2	4	5	5	6	8

j i

count = ~~0~~ ~~1~~ ~~2~~ ~~3~~ 4

sum = ~~10~~ ~~8~~ ~~9~~ 10