- Given an array of numbers that contains [1...n] inclusive w/ n+1 values.

We know there are at least one duplicate because:

[1, 2, 3, x]   x has to be a value [1...n]
all which already exist.

n=3

Input: [1 3 4 2 2]

Output: 2

Approach #1

1.) Sort array

2.) Compare each i w/ i-1 to see if identical
At least 1 will match, return that match

Time: O(nlogn)   Space: generally known to be O(n)
for sorting

We are also given restriction: we cannot modify array
and can only use constant
space.

[1 3 4 2 2]

Potential Sum: $\frac{n(n+1)}{2} = \frac{4(4+1)}{2} = 10$

Actual Sum: 12

Answer: Actual Sum - Potential Sum

[1 1 3 4 2]

$PS = \frac{4(4+1)}{2} = 10$

$AS = 11$

$Ans = 11 - 10$

This only works if ALL values are unique except duplicate

Input: [2 2 2 2]

Output: 2

Idk how to not manipulate array but I can keep O(n) time and O(1) space by manipulating.

Input:	[1 3 4 2 2]	
	[1 -3 4 2 2]	[-1 3 4 2 2]
	[1 -3 4 -2 2]	[-1 3 -4 2 2]
	[1 -3 4 -2 -2]	[-1 3 -4 -2 2]
	[1 -3 -4 -2 -2]	[-1 -3 -4 -2 2]
	[1 -3 -4 -2 -2]	

First idx that is already (-) is answer