# 4 $\Omega(n)$ time

$\Omega(1)$ Space

Can modify array in-place

3, 4, -1, 1 → 2

Partition so neg. on left, pos on right (pivot = 0)

Just example

→ -1 ¦ 3, 4, 1 Counting sort

CANT Do!

Counting Sort uses extra Space

-1, 1, 3, 4

↑

Start

```
x = maintain diff to prev = 0
while( in array){
  if  x > 1
      return array[prev index] + 1

   else
      goto next

  }
return 1;
```

$-1, \underbrace{1, 3, 4} \rightarrow 2$ 3 items in our partition

no matter what, solution is

either 1, 2, 3, 4

partition size + 1 possibilities

Since only pos # we can use

neg. ~~index~~ array element trick

```
                                S
                                ||
for (i=partition index to  array.size)
                           - partition index + 1
    If (array(i)  ≤  S and array(i) ≥ 1)
        index = |array[i]| - 1 ~~[illegible]~~ + partition index
        If ( array[index] > 0 )
            array[index] ~~*~~ *= -1
```

$-1, 4, 3, 1$ (actual partitioned positions)

~~-1, 4, -3, -1~~ ~~(index of 4) + 1 + partitionIndex~~

$-1, -4, 3, -1$ → (index of 3) − partitionIndex + 1

2 − 1 + 1

= 2

```
                                  S
                                  ||
for (i = partition index to  array size ){
    if (array[ i ] ≥ 0 ) return i - partition index + 1
}
return S - partition index + 1
```