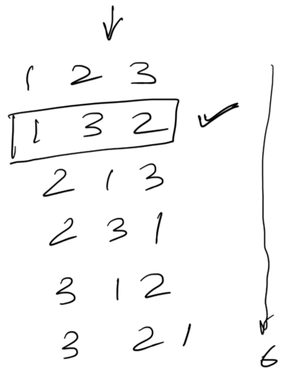
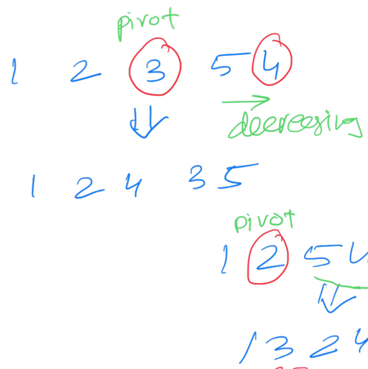
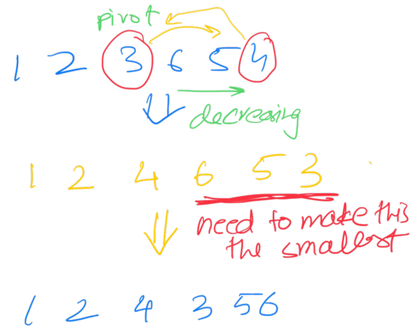
Next permutation

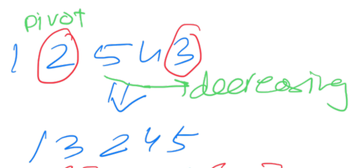
A = [1, 2, 3] return lexicographically next in same array only

O(1) SC & O(n) TC









(1) pivot ⇒ A[i] < A[i+1]

(2) find the right most element > pivot
swap (RME, pivot)

(3) reverse (pivot+1) to n-1
dec → inc

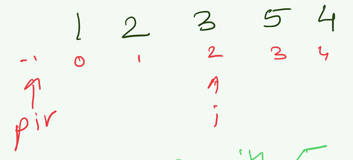
Pseudo code

```
(1) piv = -1
    for (i = n-2 ; i >= 0 ; i--) {
        if (A[i] < A[i+1]) {
            piv = i
            break
        }
    }
    if (piv == -1) {
        reverseArray()
        return
    }
(2) for (i = n-1 ; i > pivot ; i--) {
        if (A[i] > A[pivot]) {
            swap (A[i], A[pivot])
            break
        }
    }
(3) i = piv+1    j = n-1
    while (i <= j) {
        swap (A[i], A[j])
        i++
        j--
    }
```

Dry run

(1)

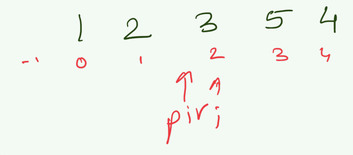
compare 5 with 4



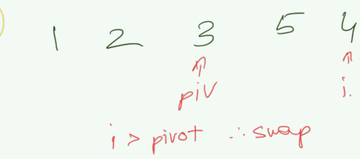
compare 3 with 5

3 < 5

piv = i

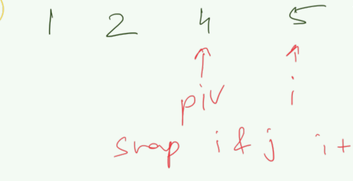


(2)

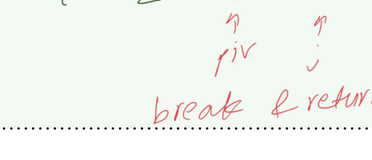


i > pivot ∴ swap

(3)



swap i & j i++ & j--



break & return