

Doubt

```

for (i=0; i<2n; ++i) {
    j=i
    while (j>0)
        j--
}

```

i	j: [i, 1]	iteration
0	[0, 1]	0
1	[1, 1]	1
2	[2, 1]	2
⋮	⋮	⋮
2 ⁿ⁻¹		2 ⁿ⁻¹

$$\cancel{1} + 1 + 2 + \dots + 2^{n-1}$$

$$1 + 2 + \dots + 2^{n-1}$$

$$\frac{2^{n-1} (2^{n-1} + 1)}{2}$$

$$= \frac{4^{n-1} + 2^{n-1}}{2}$$

$$\boxed{4^n}$$

$$\frac{4^n}{8}$$

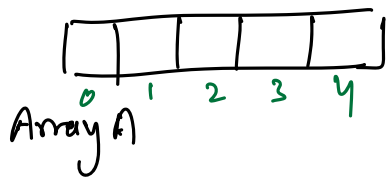
$$\rightarrow \boxed{\frac{2^{n-1}}{2}}$$

Introduction to Arrays

Array?

↳ contiguous list of homogenous data

↳ size is fixed. (static size)



→ index to access a value in array

for an array of size N : 0 to $N-1$ indices
 $0, 1, 2, 3, \dots, N-2, N-1$

$a[N]$

$a[i] = ?$

$TC = O(1)$

To access all the elements of array, it will take $O(N)$ time.

$a[10] \rightarrow a[0], a[1], \dots, a[9], a[10], a[11], \dots$
 $a[1]$

Traverse in Array

$a[n]$

iterations = N

for($i=0$; $i<n$; $++i$) {

TC = $O(N)$

$a[i]$

}

Question 1

Given an array of size N . find count of elements which has at least one greater element than itself.

eg 1 4 5 3 -1 5 4 ans = 5

eg 2 5 1 4 8 0 8 1 3 8 ans = 7

1. find the max element
2. count occurrences of max element
3. ans = $N - \text{count}$

$mx = a[0]$

```
for (i=1; i<n; ++i) {  
    if (a[i] > mx)  
        mx = a[i]  
}
```

$N-1$
iterations

count = 0

```
for (i=0; i<n; ++i) {  
    if (a[i] == mx)  
        ++count  
}
```

N
iterations

iterations = $2N-1$

$T = O(N)$

$SC = O(1)$

ans = $N - \text{count}$

HW: Solve in one
for loop.

If $mx = 0$

$a[3] = -3 \quad -1 \quad -2$

In case of all -ive values, $mx=0$ will not work.

Question 2

Given an array of N elements, find count of pair (i, j) where i, j are indices such that

$$a[i] + a[j] = \textcircled{K} \rightarrow \text{given } i < j$$

eg $a[6] = \{ \underset{0}{1} \quad \underset{1}{5} \quad \underset{2}{6} \quad \underset{3}{2} \quad \underset{4}{4} \quad \underset{5}{3} \}$ $K=7$

$(0, 2)$

$(1, 3)$

$(4, 5)$

$\boxed{\text{ans} = 3}$

$a[10] = \underset{0}{3} \quad \underset{1}{5} \quad \underset{2}{2} \quad \underset{3}{1} \quad \underset{4}{-3} \quad \underset{5}{7} \quad \underset{6}{8} \quad \underset{7}{15} \quad \underset{8}{6} \quad \underset{9}{13}$

$K=10$

$(0, 5)$

$(2, 6)$

$(4, 9)$

$\boxed{\text{ans} = 3}$

$a[4]$

$\{0, 1, 2, 3\}$

	$i \quad j$		
$(0, 0)$	$(0, 1)$	$(0, 2)$	$(0, 3)$
$(1, 0)$	$(1, 1)$	$(1, 2)$	$(1, 3)$
$(2, 0)$	$(2, 1)$	$(2, 2)$	$(2, 3)$
$(3, 0)$	$(3, 1)$	$(3, 2)$	$(3, 3)$

ans = 0

```
for (i = 0; i < n - 1; ++i) {  
    for (j = i + 1; j < n; ++j) {  
        // (i, j) is valid  
        if (a[i] + a[j] == k)  
            ++ans  
    }  
}
```

print(ans)

i	j: [i+1, n-1]	iterations
0	[1, n-1]	n-1 ⁺
1	[2, n-1]	n-2 ^f
2		n-3 ⁺
⋮		⋮
⋮		⋮
⋮		⋮
⋮		⋮
n-2	[n-1, n-1]	1

$$= n-1 + n-2 + \dots + 1$$

$$\Rightarrow 1 + 2 + \dots + n-1$$

$$\text{iterations} = \frac{(n-1)(n-1+1)}{2}$$

$$= \frac{n(n-1)}{2}$$

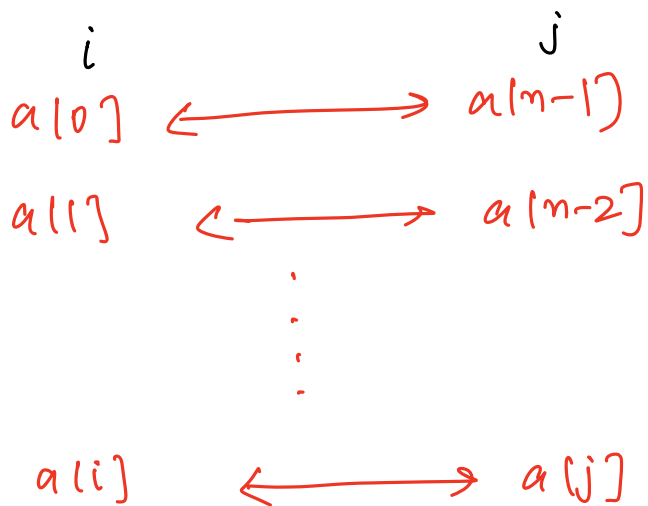
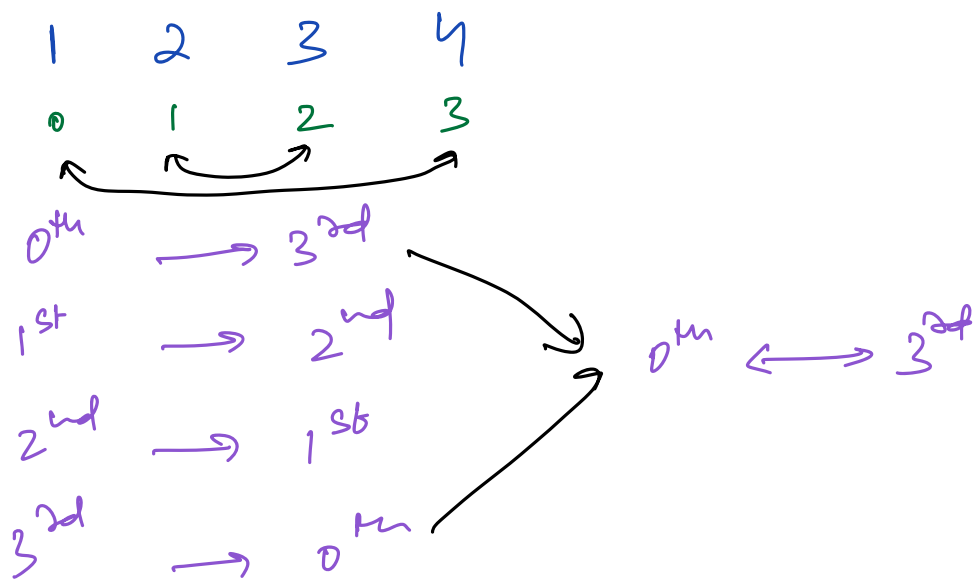
$$\text{TC: } O(N^2)$$

$$\text{SC: } O(1)$$

Question 3

Given an array of size N. Reverse the array without using extra space.

eg $\{1, 2, 3, 4\} \xrightarrow{\text{reverse}} \{4, 3, 2, 1\}$



$$0 + n - 1 = n - 1$$

$$1 + n - 2 = n - 1$$

$$i + j = n - 1$$

$$j = n - i - 1$$

for($i=0$; $i < n$; $++i$) {
 swap($a[i]$, $a[n-i-1]$)
 }

} X NOT WORKS

$a[4] = 1 \ 2 \ 3 \ 4$

$i=0$	$j=4-0-1=3$	4	2	3	1
$i=1$	$j=2$	4	3	2	1
$i=2$	1	4	2	3	1
$i=3$	0	1	2	3	4

```

i=0, j=n-1
while (i < j) {
    swap(a[i], a[j])
    i++, j--
}

```

WORKS

iterations = $n/2$

TC: $O(N)$

SC: $O(1)$

What if we have to reverse only some part of array. [start, end]

arr: 1 2 3 4 5 6 7

0 1 2 3 4 5 6

↑ start ↑ end

⇓

1 5 4 3 2 6 7

i = start, j = end

```

while (i < j) {
    swap(a[i], a[j])
    ++i, j--
}

```

TC: $O(\text{end} - \text{start})$

: $O(N)$ → worst case

SC: $O(1)$

BREAK: 8:16 - 8:26

Question 4

Given an array of size N . Rotate the array K times clockwise. $K < N$

eg

	1	2	3	4	5
	0	1	2	3	4

$K=1$

5	1	2	3	4
---	---	---	---	---

$K=2$

4	5	1	2	3
---	---	---	---	---

$K=3$

3	4	5	1	2
0	1	2	3	4

$K=3$

$$a[0] = 1 \xrightarrow{+3} a[3] = 1$$

$$a[1] = 2 \xrightarrow{+3} a[4] = 2$$

$$a[2] = 3 \xrightarrow{-2} a[0] = 3$$

$$a[3] = 4 \xrightarrow{-2} a[1] = 4$$

$$a[4] = 5 \xrightarrow{-2} a[2] = 5$$

temp[n]
for ($i=0; i < n; ++i$) {

$$j = (i + K) \% n$$

$$\text{temp}[j] = a[i] \quad [a[j] = a[i]]$$

}

↳ len of value

for ($i=0; i < n; ++i$) {

$$a[i] = \text{temp}[i]$$

}

$$(0+3) \% 5 = 3$$

$$(1+3) \% 5 = 4$$

$$(2+3) \% 5 = 0$$

$$(3+3) \% 5 = 1$$

$$(4+3) \% 5 = 2$$

$$TC: O(N)$$

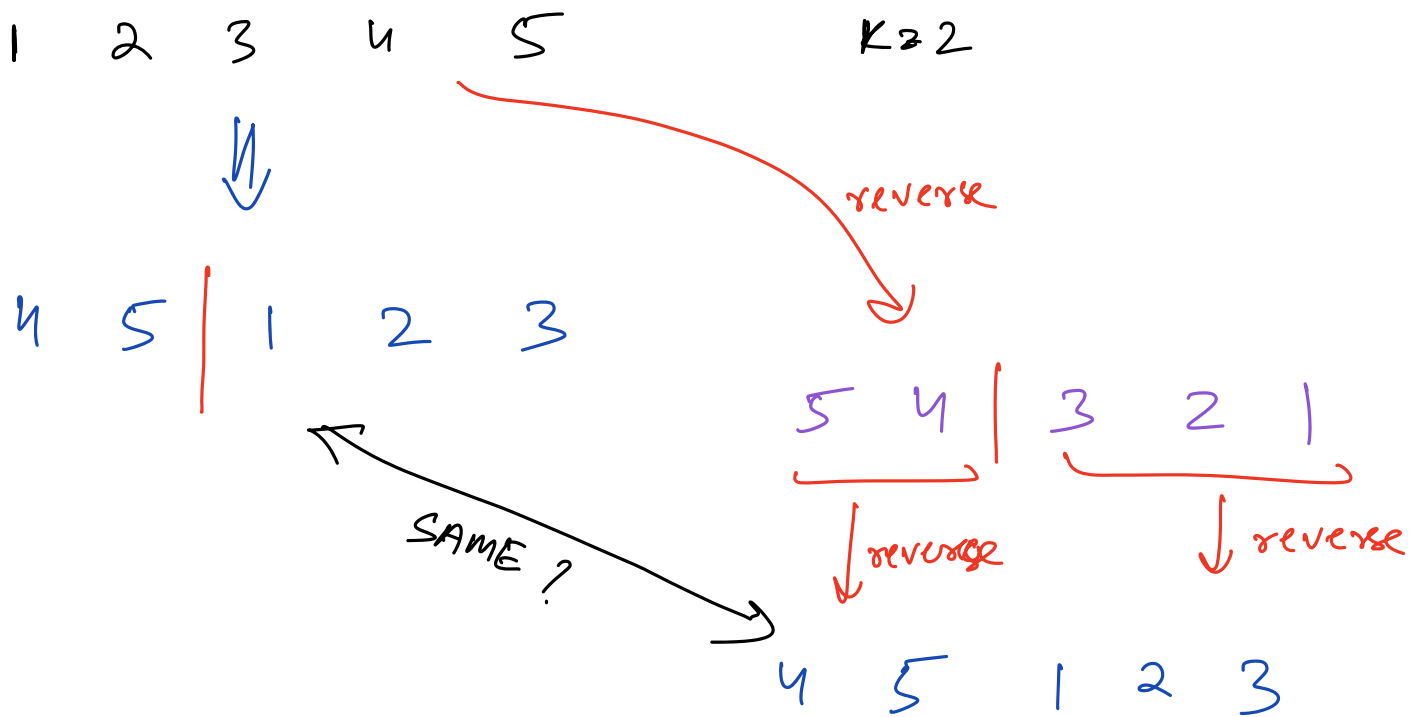
$$SC: O(N)$$

If we don't use extra space, we can shift array one by one in K iterations.

```
for (i=0; i<K; ++i) {  
    temp = a[n-1]  
    for (j=n-1; j>0; --j) {  
        a[j] = a[j-1]  
    }  
    a[0] = temp  
}
```

TC: $O(K \times N)$

SC: $O(1)$



To shift elements K times clockwise \Rightarrow
iterations

1. reverse $(0, N-1) \Rightarrow N/2$
2. reverse $(0, K-1) \Rightarrow K/2$
3. reverse $(K, N-1) \Rightarrow \frac{N-K}{2}$

$$\text{total iterations} = \frac{N}{2} + \frac{K}{2} + \frac{N-K}{2} = N$$

TC: $O(N)$

SL: $O(1)$

reverse $(a, 0, n-1)$
reverse $(a, 0, K-1)$
reverse $(a, K, n-1)$

```
void reverse(a[], start, end) {  
    i = start, j = end  
    while (i < j) {  
        swap(a[i], a[j])  
        i++, j--  
    }  
}
```

Dynamic Array : Array with variable size

C++	Java	Python	JS/Ruby
vector	arraylist	list	array

append()/add()/insert()/push-back()

HW: Understand dynamic array in your language.

Doubt

```
for (i = N; i > 0; i /= 2) {
    for (j = 0; j < i; ++j) {}
}
```

i	j : [0, i-1]	
N	[0, N-1]	N
N/2	[0, N/2-1]	N/2
N/4		N/4
⋮		⋮
1		1

1 + 2 + 4 + ... + N

$2^0, 2^1, 2^2, \dots, N$

$k = \log_2 N$

$$1 \left(\frac{2^{\log_2 N} - 1}{2 - 1} \right) = 2^{\log_2 N} - 1 = N - 1 \quad \text{DN}$$

$$\log_a a^c = c$$

```

i = 1
while (i < n) {
    x = i;
    while (x-- > 0) {}
    ++i
}

```

i	x = [i, 0]	
1	[1, 0]	1
2		2
⋮		3
⋮		⋮
⋮		⋮
N-1		N-1

$$1 + 2 + 3 + \dots + N-1$$

$$\frac{N(N-1)}{2}$$

$$\approx O(N^2)$$