

## Agenda

- 1) Basics [ 5-7 min ]
- 2) Questions  
4-5 problems .
- 3) TODOS.

Basics:

int A[] = { 3, 2, 8, 9, 7 }

1st element index: 0

last element index: 4

Generalize for n ele

1st ele index: 0

last ele index: n-1

To access any element, it takes O(1) time

```
for(i=0; i<n; i++) {  
    print(A[i])  
}
```

$\Rightarrow T.C \ O(n)$

Ques 1. Given N array elements, count no. of elements having atleast 1 element greater than itself.

$$A[1] = \left\{ \begin{matrix} 0 \\ 3, -2, 6, \frac{3}{8}, \frac{4}{4}, \frac{5}{8}, \frac{6}{5} \end{matrix} \right\}$$

+1 +1 +1 0 +1 0 +1

ans: 5

Observation:

Ques. which ele won't have any ele greater than itself?  
 Ans. MAX ELEMENT.

# Rest of the elements will have atleast 1 ele greater than themselves

Ans: #total - ① ↴ subtracted max ele.  
 WRONG?

There can be more than 1 max element.

Ans: #total - #count of max-ele.

max-ele =  $A[0]/-\infty$   
 for( $i=0; i < n; i++$ ) {  
 | if ( $A[i] > \text{max-ele}$ ) {  
 | | max-ele =  $A[i]$   
 | }  
 }  
 }

$T_C$   
 $O(n)$ .

count\_max = 0  
 for( $i=0; i < n; i++$ ) {  
 | if ( $A[i] == \text{max-ele}$ ) {  
 | | count\_max ++  
 | }  
 }  
 return  $n - \text{count\_max}$ ;

# TODO:  
 Execute this in a  
 single iteration

Ques 2. check if there exists a pair  $(i, j)$  s.t  $A[i] + A[j] = K$

Note:  $i \neq j$

No inbuilt lib can be used!  
No extra space!

$$A[] = \{ 3^0, -2^1, 1^2, 4^3, 3^4, 6^5, 8^6 \}$$

$$\begin{array}{ll} K=10 & 4+6 \\ \equiv & 3 \quad 5 \end{array} \text{ True}$$

$$\begin{array}{ll} K=8 & \text{Not possible} \end{array}$$

Hint: Read question carefully  
make observations  
Logic.

Pair!

check each pair  
if there sum is  $K$   
then return true  
return false.

let's create all the pairs!

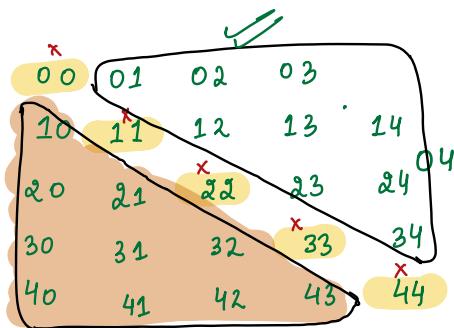
5 ele

0 1 2 3 4

```
for(i=0; i<n; i++) {  
    for(j=0; j<n; j++) {  
        if(A[i] + A[j] == K) {  
            return true  
        }  
    }  
}
```

return false.

}



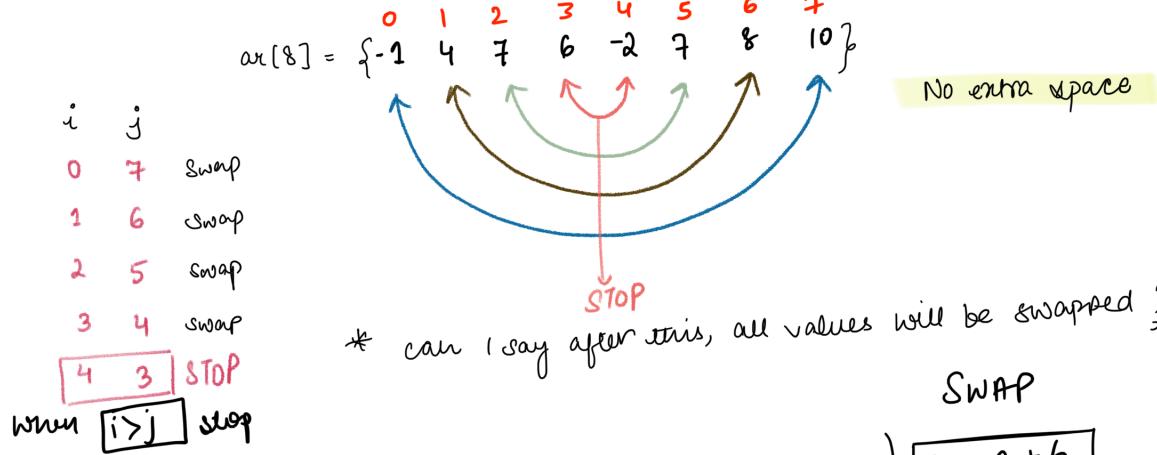
```
for(i=0; i<n; i++) {  
    for(j=i+1; j<n; j++) {  
        if(A[i] + A[j] == K) {  
            return true  
        }  
    }  
}
```

return false.

$T.C \rightarrow O(n^2)$   
 $S.C \rightarrow O(1)$

Que 3 Given an array, reverse entire array

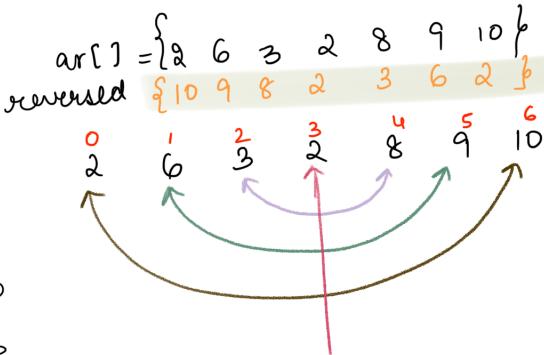
ar[8] = { -1 4 7 6 -2 7 8 10 }  
After reversing      { 10 8 7 -2 6 7 4 -1 }  
 \* The original array should change



another eg

i   j  
 0   6 swap  
 1   5 swap  
 2   4 swap  
 3   3 stop

when  $i = j$  stop



$$\begin{aligned} a &= a + b \\ b &= a - b \\ a &= a - b \end{aligned}$$

XOR

$$\begin{aligned} a &= a \wedge b \\ b &= a \wedge b \vee a \wedge b \wedge \neg b \\ a &= a \wedge b \vee a \wedge b \wedge \neg b \end{aligned}$$

Swap(a, b)

look at i's & j's &  
 tell how are we updating  
 them?

```
void swap(A, n){  

    i=0, j=n-1  

    while(i < j){  

        swap(ar[i], ar[j])  

        i++  

        j--  

    }  

}
```

will execute  
 this only once  
 - NB put inside  
 so, put inside  
 while loop

Iterations? — N/2

TC → O(N)

SC → O(1)

Ques. Given arr of N ele, rotate array in clockwise direction by K times. [No extra space]

arr[7] : { 0 1 2 3 4 5 6 }  
 arr[7] : { 3 -2 1 4 6 9 8 }  
 Rotate 1 : { 8 3 -2 1 4 6 }  
 Rotate 2 : { 9 8 3 -2 1 4 }  
 Rotate 3 : { 6 9 8 3 -2 1 }  
 Rotate 4 : { 4 6 9 8 3 -2 1 }

arr[7] : 3 -2 1 4 6 9 8  
 Rotate by 4 : 4 6 9 8 3 -2 1

↳ last 4 ele, they are at start  
 first 3 ele, " " " end

arr[10] : -2 3 1 4 6 2 8 7 9 3  
 Rotate 3 : 7 9 3 -2 3 1 4 6 2 8

arr[13] : a<sub>0</sub> a<sub>1</sub> a<sub>2</sub> a<sub>3</sub> a<sub>4</sub> a<sub>5</sub> a<sub>6</sub> a<sub>7</sub> a<sub>8</sub> a<sub>9</sub> a<sub>10</sub> a<sub>11</sub> a<sub>12</sub>  
 Rotate 5 : a<sub>8</sub> a<sub>9</sub> a<sub>10</sub> a<sub>11</sub> a<sub>12</sub> a<sub>0</sub> a<sub>1</sub> a<sub>2</sub> a<sub>3</sub> a<sub>4</sub> a<sub>5</sub> a<sub>6</sub> a<sub>7</sub>

### Observation

Reverse : a<sub>0</sub> a<sub>1</sub> a<sub>2</sub> a<sub>3</sub> a<sub>4</sub> a<sub>5</sub> a<sub>6</sub> a<sub>7</sub> a<sub>8</sub> a<sub>9</sub> a<sub>10</sub> a<sub>11</sub> a<sub>12</sub>  
 a<sub>12</sub> a<sub>11</sub> a<sub>10</sub> a<sub>9</sub> a<sub>8</sub> a<sub>7</sub> a<sub>6</sub> a<sub>5</sub> a<sub>4</sub> a<sub>3</sub> a<sub>2</sub> a<sub>1</sub> a<sub>0</sub>

Observe rotated & reversed array  
 Is there any pattern?  
 Can u get rotated array using the reversed array?  
 They are same but only the order differs.

reverse : a<sub>8</sub> a<sub>9</sub> a<sub>10</sub> a<sub>11</sub> a<sub>12</sub> a<sub>0</sub> a<sub>1</sub> a<sub>2</sub> a<sub>3</sub> a<sub>4</sub> a<sub>5</sub> a<sub>6</sub> a<sub>7</sub>

↓ reverse

## Pseudocode

```

reverse(ar, 0, N-1)      → N/2
// reverse first K ele
reverse(ar, 0, K-1)      → K/2
// reverse last N-K ele
reverse(ar, K, N-1)      →  $\frac{N-K}{2}$ 

```

} Total  
 $O(N)$  → TC  
 $O(1)$  → SC

Note:  $K > N$

reverse(ar, 0, N-1) ✓  
 reverse(ar, 0, K-1) ✗ [index out of bounds]

arr[6]: a <sub>0</sub> a <sub>1</sub> a <sub>2</sub> a <sub>3</sub> a <sub>4</sub> a <sub>5</sub>	Rotate 0	Rotate 1	Rotate 2	Rotate 3	Rotate 4	Rotate 5	Rotate 6	Rotate 7	Rotate 8	Rotate 9	Rotate 10
a <sub>0</sub> a <sub>1</sub> a <sub>2</sub> a <sub>3</sub> a <sub>4</sub> a <sub>5</sub>	a <sub>0</sub> a <sub>1</sub> a <sub>2</sub> a <sub>3</sub> a <sub>4</sub> a <sub>5</sub>	a <sub>5</sub> a <sub>0</sub> a <sub>1</sub> a <sub>2</sub> a <sub>3</sub> a <sub>4</sub>	a <sub>4</sub> a <sub>5</sub> a <sub>0</sub> a <sub>1</sub> a <sub>2</sub> a <sub>3</sub>	a <sub>3</sub> a <sub>4</sub> a <sub>5</sub> a <sub>0</sub> a <sub>1</sub> a <sub>2</sub>	a <sub>2</sub> a <sub>3</sub> a <sub>4</sub> a <sub>5</sub> a <sub>0</sub> a <sub>1</sub>	a <sub>1</sub> a <sub>2</sub> a <sub>3</sub> a <sub>4</sub> a <sub>5</sub> a <sub>0</sub>	a <sub>0</sub> a <sub>1</sub> a <sub>2</sub> a <sub>3</sub> a <sub>4</sub> a <sub>5</sub>				
	0	1	2	3	4	5	6	7	8	9	10
							12	13	14	15	16
							18	19	20	21	22
											23
											$a_1 \cdot 6 = 3$

Same

rotate 0

rotate 1

rotate 2

rotate 3

rotate 4

rotate 5

rotate 6

rotate 7

rotate 8

rotate 9

if  $K > N$

$K = K \% N$

## Dynamic Arrays

C++	Python	Java	C#	JS	C
vector	list	ArrayList	ArrayList	array	move to another long.