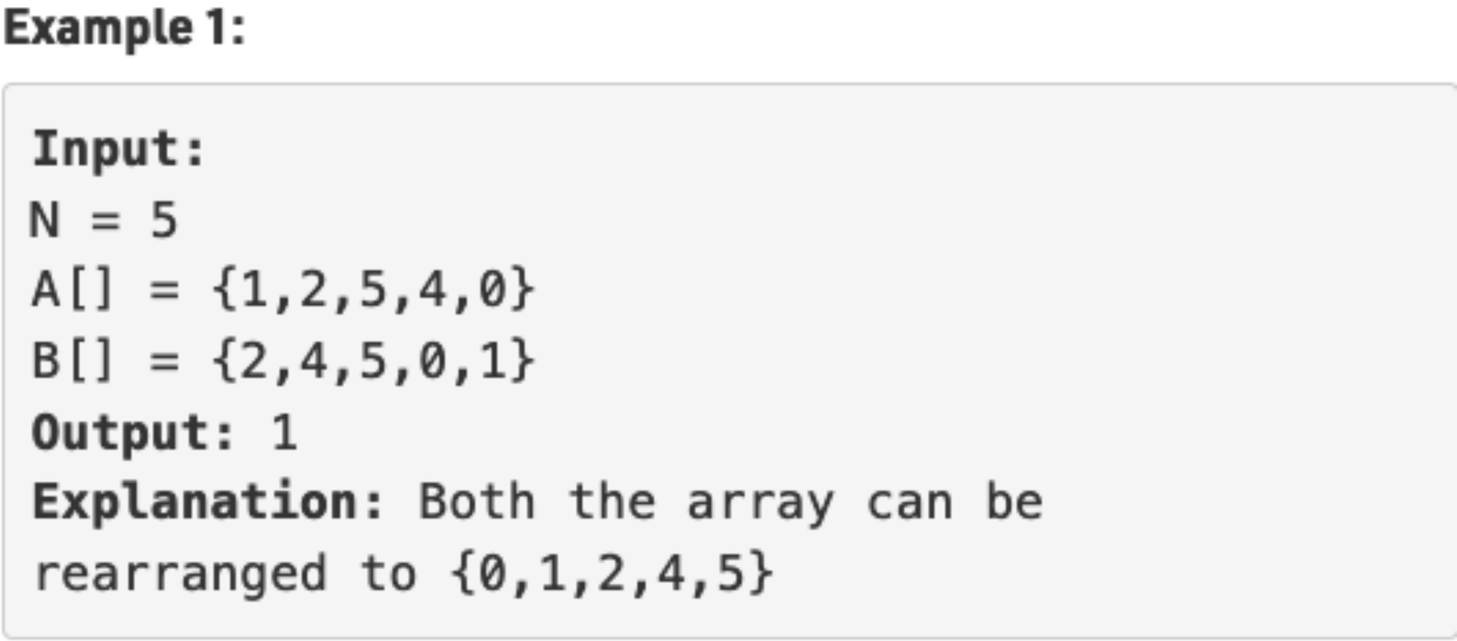
HASHING DATA STRUCTURE

- **Check if two arrays are equal or not**
- **Given two arrays A and B of equal size N**
- 

Example 1:

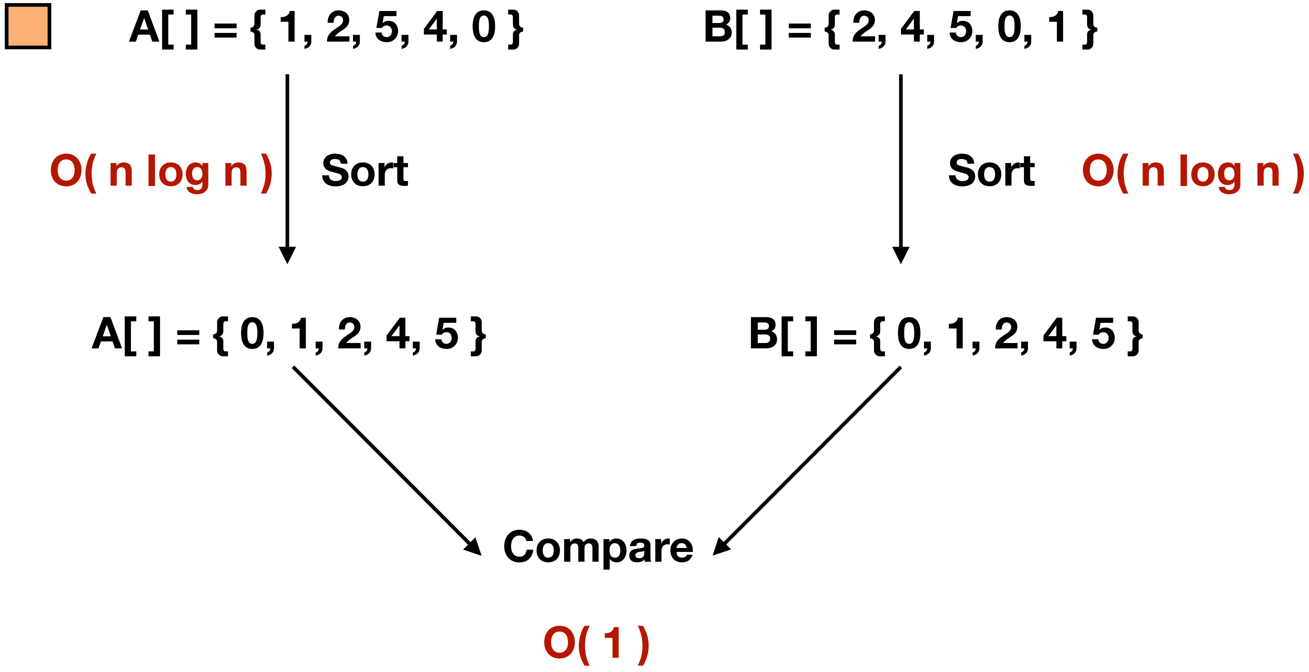
```
Input:
N = 5
A[] = {1,2,5,4,0}
B[] = {2,4,5,0,1}
Output: 1
Explanation: Both the array can be
rearranged to {0,1,2,4,5}
```

Hello World

HASHING DATA STRUCTURE

- Check if two arrays are equal or not

- Now, our general Approach is like sort and Compare

- 

A[ ] = { 1, 2, 5, 4, 0 }

B[ ] = { 2, 4, 5, 0, 1 }

O(n log n) Sort

Sort O(n log n)

A[ ] = { 0, 1, 2, 4, 5 }

B[ ] = { 0, 1, 2, 4, 5 }

Compare

O(1)

HASHING DATA STRUCTURE

- **Logics** **Iterate all elements in unordered_map**

- **A[ ] = { 1, 2, 5, 4, 0 , 2}** **B[ ] = { 2, 4, 5, 0, 1 }**

- **A[ ] = { 1, 2, 5, 4, 0 , 2}**

0 -> 1

1 -> 1

2 -> 2

4 -> 1

5 -> 1

✗ **because, in question it is mention that,**
Number of elements are equal

Hello World

HASHING DATA STRUCTURE

- Logics

Iterate all elements in unordered_map

- A[ ] = { 1, 2, 5, 4, 0 , 2}

B[ ] = { 2, 4, 5, 0, 1, 2 }

- A[ ] = { 1, 2, 5, 4, 0 , 2}

umap =

0 -> 1

1 -> 1

2 -> 2

4 -> 1

5 -> 1

Now, Iterate elements in B[ ]

And if we found element then Decrease it's count

I = 0

Key = 2

Hello World

HASHING DATA STRUCTURE

- Logics

Iterate all elements in unordered_map

- A[ ] = { 1, 2, 5, 4, 0 , 2}

B[ ] = { 2, 4, 5, 0, 1, 2 }

- A[ ] = { 1, 2, 5, 4, 0 , 2}

umap =

0 -> 1
1 -> 1
2 -> 1
4 -> 1
5 -> 1

Now, Iterate elements in B[ ]
And if we found element then Decrease it's count

I = 0

Key = 2

Hello World

HASHING DATA STRUCTURE

- Logics

Iterate all elements in unordered_map

- A[ ] = { 1, 2, 5, 4, 0 , 2}

B[ ] = { 2, 4, 5, 0, 1, 2 }

- A[ ] = { 1, 2, 5, 4, 0 , 2}

Now, Iterate elements in B[ ]
And if we found element then Decrease it’s count

umap =

0 -> 1
1 -> 1
2 -> 1
4 -> 1
5 -> 1

I = 1

Key = 4

Hello World

HASHING DATA STRUCTURE

- Logics

Iterate all elements in unordered_map

- A[ ] = { 1, 2, 5, 4, 0 , 2}

B[ ] = { 2, 4, 5, 0, 1, 2 }

- A[ ] = { 1, 2, 5, 4, 0 , 2}

umap =

0 -> 1
1 -> 1
2 -> 1
4 -> 0
5 -> 1

Now, Iterate elements in B[ ]
And if we found element then Decrease it's count

I = 1

Key = 4

Hello World

HASHING DATA STRUCTURE

- Logics

Iterate all elements in unordered_map

- A[ ] = { 1, 2, 5, 4, 0 , 2}

B[ ] = { 2, 4, 5, 0, 1, 2 }

- A[ ] = { 1, 2, 5, 4, 0 , 2}

umap =

0 -> 1

1 -> 1

2 -> 1

4 -> 0

5 -> 1

Now, Iterate elements in B[ ]
And if we found element then Decrease it's count

I = 2

Key = 5

Hello World

HASHING DATA STRUCTURE

- Logics

Iterate all elements in unordered_map

- A[ ] = { 1, 2, 5, 4, 0 , 2}

B[ ] = { 2, 4, 5, 0, 1, 2 }

- A[ ] = { 1, 2, 5, 4, 0 , 2}

umap =

0 -> 1

1 -> 1

2 -> 1

4 -> 0

5 -> 0

Now, Iterate elements in B[ ]
And if we found element then Decrease it’s count

I = 2

Key = 5

HASHING DATA STRUCTURE

- Logics

Iterate all elements in unordered_map

- A[ ] = { 1, 2, 5, 4, 0 , 2}

B[ ] = { 2, 4, 5, 0, 1, 2 }

- A[ ] = { 1, 2, 5, 4, 0 , 2}

umap =

0 -> 1

1 -> 1

2 -> 1

4 -> 0

5 -> 0

Now, Iterate elements in B[ ]

And if we found element then Decrease it's count

I = 3

Key = 0

Hello World

HASHING DATA STRUCTURE

- Logics

Iterate all elements in unordered_map

- A[ ] = { 1, 2, 5, 4, 0 , 2}

B[ ] = { 2, 4, 5, 0, 1, 2 }

- A[ ] = { 1, 2, 5, 4, 0 , 2}

Now, Iterate elements in B[ ]
And if we found element then Decrease it’s count

umap =

0 -> 0

1 -> 1

2 -> 1

4 -> 0

5 -> 0

I = 3

Key = 0

Hello World

HASHING DATA STRUCTURE

- Logics

Iterate all elements in unordered_map

- A[ ] = { 1, 2, 5, 4, 0 , 2}

B[ ] = { 2, 4, 5, 0, 1, 2 }

- A[ ] = { 1, 2, 5, 4, 0 , 2}

umap =

0 -> 0

1 -> 1

2 -> 1

4 -> 0

5 -> 0

Now, Iterate elements in B[ ]

And if we found element then Decrease it's count

I = 4

Key = 1

HASHING DATA STRUCTURE

- Logics

Iterate all elements in unordered_map

- A[ ] = { 1, 2, 5, 4, 0 , 2}

B[ ] = { 2, 4, 5, 0, 1, 2 }

- A[ ] = { 1, 2, 5, 4, 0 , 2}

umap =

0 -> 0

1 -> 0

2 -> 1

4 -> 0

5 -> 0

Now, Iterate elements in B[ ]
And if we found element then Decrease it's count

I = 4

Key = 1

Hello World

HASHING DATA STRUCTURE

- **Logics** **Iterate all elements in unordered_map**
- **A[ ] = { 1, 2, 5, 4, 0 , 2}** **B[ ] = { 2, 4, 5, 0, 1, 2 }**
- **A[ ] = { 1, 2, 5, 4, 0 , 2}**

umap =

0 -> 0

1 -> 0

2 -> 1

4 -> 0

5 -> 0

Now, Iterate elements in B[ ]
And if we found element then Decrease it's count

I = 5 **Key = 2**

Hello World

HASHING DATA STRUCTURE

- Logics

Iterate all elements in unordered_map

- A[ ] = { 1, 2, 5, 4, 0 , 2}

B[ ] = { 2, 4, 5, 0, 1, 2 }

- A[ ] = { 1, 2, 5, 4, 0 , 2}

umap =

0 -> 0

1 -> 0

2 -> 0

4 -> 0

5 -> 0

Now, Iterate elements in B[ ]

And if we found element then Decrease it's count

I = 5 Key = 2

✓ This is the right method

Hello World