

## Today's Content

1. Re-arrange arr[]
2. Count frequency of all elements.

Few Basics:

Say  $a \geq 0$ :  $K \leq a < M$ :

1.  $a \% M = a$

$$5 \% 7 = 5$$

$$6 \% 10 = 6$$

2.  $a / M = 0$

$$6 / 10 = 0$$

$$3 / 5 = 0$$

Modular:

Given an  $ar[N]$  re-arrange array value, such that  $ar[i] = ar[ar[i]]$

Note: all array elements are unique.

Constraints:

$$1 \leq N \leq 10^5$$

$$0 \leq ar[i] < N$$

Ex,  $ar[N] = \{ 6 \quad 4 \quad 2 \quad \underline{1} \quad \underline{7} \quad \underline{0} \quad \underline{3} \quad \underline{5} \}$

$ar[i] = ar[ar[i]]$

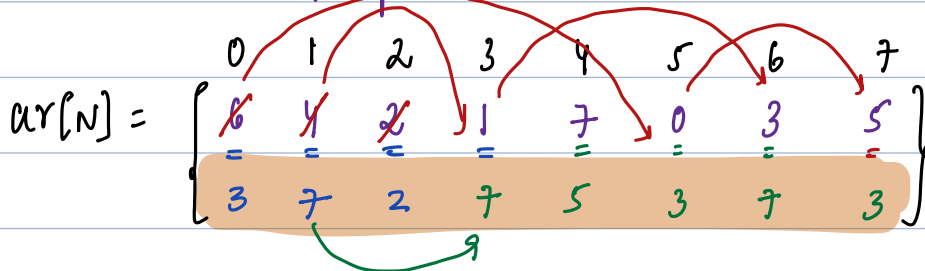
$\{ 3 \quad 7 \quad 2 \quad 4 \quad 5 \quad 6 \quad 1 \quad 0 \}$

Idea1: Using Extra Space  $T.C: O(N+N) = O(N)$   $S.C: O(N)$   
 $int b[N];$

$i=0; i < N; i++ \{$   
 $\quad b[i] = ar[ar[i]]$   
 $\}$

$i=0; i < N; i++ \{$   
 $\quad ar[i] = b[i]$   
 $\}$

Idea2: Iterate on  $ar[i]$  & update  $ar[i] = ar[ar[i]]$ .



Issue: If we update array we lose old data.

Con: We should store old & new data at same place?

#idea2:

#hrs Since Bing Bang	Days	Time
100 hrs	4D	4hrs = 4am
40 hrs	1D	16hrs = 4pm
78 hrs	3D	6hrs = 6am
$n$ hrs	$n/24$	$n\%24$

obs: In  $n$ , we are able to see both days & Time

hint:  $0 \leq \text{ar}[i] < N$  : If we divide  $\%N = [0..N-1]$

Dividend = Divisor \* Quotient + Remainder

$$\text{ar}[i] : \begin{cases} \text{ar}[i]/N = \text{old} = Q \\ \text{ar}[i]\%N = \text{new} = R \end{cases}$$

$$\text{ar}[i] = N * \text{old} + \text{new}$$

$$\begin{aligned} \text{ar}[i]/N &= (N * \text{old} + \text{new})/N \\ &= (N * \text{old}/N + \text{new}/N) \\ &= (\text{old} + 0) \\ &= \text{old} \end{aligned}$$

$$\begin{aligned} \text{ar}[i]\%N &= (N * \text{old} + \text{new})\%N \\ &= ((N * \text{old})\%N + \text{new}\%N)\%N \\ &= (0 + \text{new}\%N)\%N \\ &= \text{new}\%N = \text{new} \end{aligned}$$

$$\text{ar}[i] : \begin{cases} \text{ar}[i]\%N = \text{old} = R \\ \text{ar}[i]/N = \text{new} = Q \end{cases}$$

$$\text{ar}[i] = N * \text{new} + \text{old}$$

$$\begin{aligned} \text{ar}[i]/N &= (N * \text{new} + \text{old})/N \\ &= (N * \text{new}/N + \text{old}/N) \\ &= (\text{new} + 0) \\ &= \text{new} \end{aligned}$$

$$\begin{aligned} \text{ar}[i]\%N &= (N * \text{new} + \text{old})\%N \\ &= (N * \text{new}\%N + \text{old}\%N)\%N \\ &= (0 + \text{old}\%N)\%N \\ &= (\text{old}\%N) = \text{old} \end{aligned}$$

Approach:  $ar[i] = N * old + new$

$ar[N] = \{ \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 6 & 4 & 2 & 1 & 7 & 0 & 3 & 5 \end{matrix} \}$

#Step1: Iterate on  $ar[]$  & multiply by  $N$ .

$ar[N] = \{ \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 6*8 & 4*8 & 2*8 & 1*8 & 7*8 & 0*8 & 3*8 & 5*8 \end{matrix} \}$

#Step2: Iterate on  $ar[]$  & add new value

$ar[N] = \{ \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 6*8+3 & 4*8+7 & 2*8+2 & 1*8+7 & 7*8+5 & 0*8+6 & 3*8+1 & 5*8+0 \end{matrix} \}$   
 $i$

0  $ar[0] += ar[ar[0]/8] = ar[6]/8 = 3$        $ar[0] = ar[ar[0]]$

1  $ar[1] += ar[ar[1]/8] = ar[4]/8 = 7$

2  $ar[2] += ar[ar[2]/8] = ar[2]/8 = 2$

3  $ar[3] += ar[ar[3]/8] = ar[1]/8 = 7$

i  $ar[i] += ar[ar[i]/N]/N$

#Step3: Iterate on  $ar[]$  & apply %N

$ar[N] = \{ \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 6*8+3 & 4*8+7 & 2*8+2 & 1*8+7 & 7*8+5 & 0*8+6 & 3*8+1 & 5*8+0 \end{matrix} \}$   
 $i$

$ar[N] = \{ \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 3 & 7 & 2 & 4 & 5 & 6 & 1 & 0 \end{matrix} \}$

1000

vector<int> re-arrange(vector<int> &arr) { Tc:  $O(N+N+N) = O(N)$

Sc:  $O(1)$

3

38 Given  $arr[N]$ : Print frequency of all numbers from  $[0..N-1]$

Noted: Expected SC  $O(1)$

Constraints:

$$0 \leq arr[i] < N$$

Ex:  $arr[5] = \{ 2 \quad 4 \quad 4 \quad 0 \quad 3 \}$

Output:

0: 1

1: 0

2: 1

3: 1

4: 2

Idea 1. For every element from  $[0..N-1]$ :

Iterate & calculate frequency  
print frequency.

TC:  $O(N^2)$  SC:  $O(1)$

Hint:  $0 \leq arr[i] < N$ : If  $N = [0..N-1]$

Idea: Store freq of each element & ele at same time

$$\text{Dividend} = \text{Divisor} * \text{Quotient} + \text{Remainder}$$

$$arr[i]: \begin{cases} arr[i]/N = old = q \\ arr[i]\%N = freq = r \end{cases}$$

$$arr[i] = N * old + freq$$

$$arr[i]/N = (N * old + freq)/N$$

$$= \underline{N * old / N} + \underline{freq / N}$$

$$= (old + 1) \downarrow$$

$$= \underline{old + 1} \rightarrow \text{if all } arr[i] \text{ elements are exactly same } freq = N.$$

→ We are not able to exactly extract old value.

$$ar[i] : \begin{cases} ar[i]/N = freq = 0 \\ ar[i]\%N = old = 0 \end{cases}$$

$$ar[i] = N * freq + old$$

$$\begin{aligned} ar[i]/N &= (N * freq + old)/N \\ &= ( \underline{N * freq / N} + \underline{old / N} ) \\ &= ( freq + 0 ) \\ &= freq \end{aligned}$$

$$\begin{aligned} ar[i]\%N &= (N * freq + old)\%N \\ &= ( \underline{N * freq \% N} + old \% N ) \% N \\ &= ( 0 + old \% N ) \% N \\ &= ( old \% N ) = old \end{aligned}$$

#DryRun  $ar[i] = \underline{N * freq} + \underline{old}$  ✓

	0	1	2	3	4	5	6	7
Ex: $ar[8] = \{$	2	4	4	5	2	3	5	2 $\}$

Step1: Iterate  $q$  And frequency

	0	1	2	3	4	5	6	7
Ex: $ar[8] = \{$	<u>2</u>	<u>4</u>	<u>4 + 3N</u>	<u>5 + N</u>	<u>2 + 2N</u>	<u>3 + 2N</u>	<u>5</u>	<u>2</u> $\}$
$i$								

0  $ar[ar[0]] \neq N_j$

1  $ar[ar[1]] \neq N$

2  $ar[ar[2]\%N] \neq N_j$

3  $ar[ar[3]\%N] \neq N_j$

4  $ar[ar[4]\%N] \neq N_j$

:

$i \quad \underline{ar[ar[i]\%N]} \neq N_j \rightarrow$

Step2: Iterate  $m$   $ar[i] \> N_j$

	0	1	2	3	4	5	6	7
Ex: $ar[8] = \{$	<del>2</del>	<del>4</del>	<u>4 + 3N</u>	<u>5 + N</u>	<u>2 + 2N</u>	<u>3 + 2N</u>	<del>5</del>	<del>2</del> $\}$
$ar[8] = \{$	0	0	3	1	2	2	0	0 $\}$

TODD

void printOcc(vector<int> &arr) { Tc:  $O(N)$  Sc:  $O(1)$

3