

Sorting - 1

TABLE OF CONTENTS

1. Count Sort
2. Merge Two Sorted Arrays
3. Merge Sort
4. Calculate number of pairs such that $A[i] > B[j]$
5. Inversion Count   
6. Stable, Inplace sorting



Notes

1. Freq — HashMap

K	V
-	-

2. Using hashmap and iterating using array

3. Subarray with sum = K

Prefix Sum

HashMap

Real life example :

One Plus

Same model — together
arrange ascending order of model number

$O(N)$

$$\begin{bmatrix} 1, 2, 1, 1, 3, 2, 1, 3, 2 \end{bmatrix}$$

$$\downarrow$$

$$1 \ 1 \ 1 \ 1 \quad 2 \ 2 \ 2 \quad 3 \ 3$$

1	2	3	freq map
1	3	2	
2	1	3	



Count Sort

< Question > : Find the smallest number that can be formed by rearranging the digits of the given number in an array. Return the smallest number in the form of an array.

Counting sort

arr[] → [6 3 4 2 7 2 1]
 • 1 2 2 3 4 6 7
 arr[] → [4 2 7 3 9 0]
 ↓
 0 2 3 4 7 9



harder way - hashMap

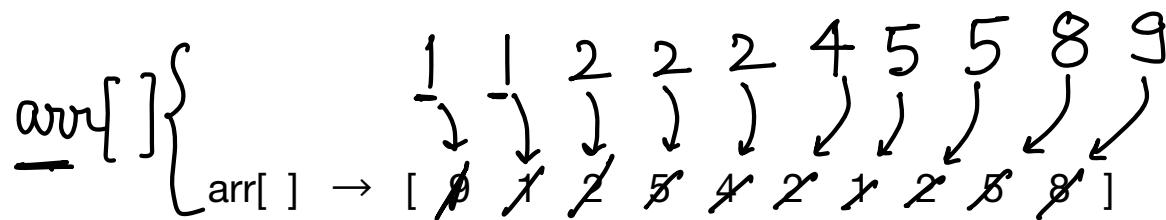
1. for 0 - 9 ... create the freq map
 2. loop on freq map — build smallest number



arr[i] → indices
 of arr[]
 arr[] → [9, 1, ?, 5, ., 4, ?, 1, 2, 5, ., 8]
0-9
 frequency array ↓
 farr → [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
 Range of array values = 0 to 9 ↑
 → farr [arr[i]] ++



frequency array ↓	$\frac{2}{X}$	$\frac{3}{X}$	$\frac{1}{X}$	$\frac{2}{X}$	$\frac{1}{X}$	$\frac{0}{X}$	$\frac{0}{X}$	$\frac{0}{X}$	$\frac{1}{X}$	$\frac{1}{X}$
farr →	0	0	0	0	0	0	0	0	0	0



< / > Code

- `int [] farr = new int[10];`

$O(N)$ { for ($i=0$ $i < \underline{\text{arr.length}}$ $i++$) {
 farr [arr[i]] ++
 }
 }

- $K=0$
`for ($j=0$ $j \leq 9$ $j++$) { //farr
 while(farr[j] != 0) {
 arr[K] = j
 K++
 farr[j] --
 }
 }`

$O(\text{Sum of all freq})$

$= O(N)$

{

TC: $O(N)$, Inplace: $O(1)$

Count Sort for large values

if you allow any integer value

- [1001, 10000, 5694, 794]

↓ Count Sort

⇒ $\frac{10^9}{10^9}$ size array

Range : 0 10

⇒ 4B

$$\frac{10^9 \times 4B}{4 \times \frac{10^9 B}{1GB}} = \underline{\underline{4GB RAM}}$$

✓ integers → $\frac{10^6}{10^6}$

Range : 0 10⁶ ⇒ $\frac{10^6 \times 4B}{10^6} = \underline{\underline{4MB}}$



Count Sort

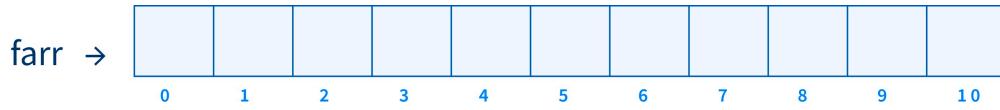
How to handle negative numbers?

1) find Range

$$\text{arr[]} \rightarrow [-2 \ 3 \ 8 \ 3 \ -2 \ 3]$$

$$\min = \underline{-2} \rightarrow \underline{2}$$

$$\max = 8$$



$$\begin{array}{cccccccccc} -2 & -1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline \end{array}$$

mapping

$$\begin{array}{ll} \text{ele} & - \min \\ -2 & \xrightarrow{-(-2)} 0 \\ \hline \end{array} \quad \begin{array}{ll} \text{abs}(\min) \\ 0 = -2 - (-2) = 0 \\ 1 = -1 - (-2) = 1 \\ 2 = 0 - (-2) = 2 \\ 3 = 1 - (-2) = 3 \\ \vdots \end{array}$$

$$\begin{array}{ll} -1 & +2 \\ \hline \end{array}$$

$$\begin{array}{ll} 0 & +2 \\ \hline \end{array}$$

$$\begin{array}{ll} 1 & +2 \\ \hline \end{array}$$

$$\begin{array}{ll} 2 & +2 \\ \hline \end{array}$$

$$\begin{array}{c} \frac{-2}{\min} \qquad \frac{\max - \min + 1}{b - a + 1} \qquad \frac{8}{\max} \\ \hline a \qquad \qquad \qquad b \end{array}$$

$$\begin{array}{l} 8 - (-2) + 1 \\ = 11 \\ \hline 0 \text{ to } 10 \\ \text{indexes} \end{array}$$



</> Code

```
{  
    find max = mx  
    find min = mn  
    int [ ] farr = new int [mx-mn+1]  
    N = arr.length  
    for ( i=0 i < N i++) {  
        farr [arr[i] - min ] ++  
    }  
}
```

Dry Run:

$$\left[\begin{array}{ccccc} -4 & \underline{5} & 6 & 2 & 3 \end{array} \right] \quad \downarrow$$

-4 2 3 5 6 (output)

$$\rightarrow mx = 6 \quad \left. \begin{matrix} \\ mn = -4 \end{matrix} \right\} = 6 - (-4) + 1$$
$$= \underline{\underline{11}}$$

$$\rightarrow \begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & \frac{1}{9} & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ -4 & -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ - & - & - & - & & & & & & & \end{matrix}$$

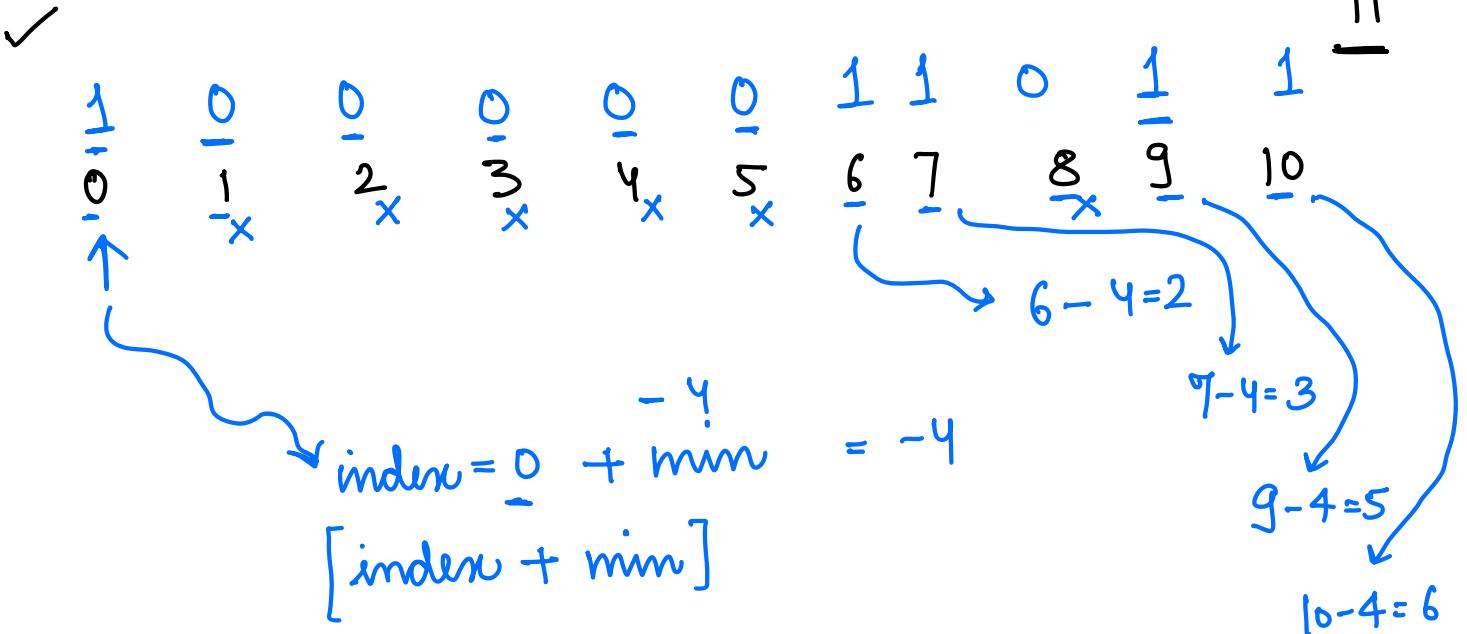
$$-4 \rightarrow \text{farr}[-4 - (-4)] = \text{farr}[0]++$$

$$5 \quad \text{farr}[5 - (-4)] = \text{farr}[9]++$$

$$6 \quad \text{farr}[6 - (-4)] = \text{farr}[10]++$$

$$2 \quad \text{farr}[2 - (-4)] = \text{farr}[6]++$$

$$3 \quad \longrightarrow \text{farr}[7]++$$



$$5 \longrightarrow -4 \quad 2 \quad 3 \quad 5 \quad 6$$

$$\underline{6 - (-4) + 1 = 11}$$

TC: $O(\text{sum of all freq})$

Code:

```

K = 0
for each index i in favor {
    while (favor[i] > 0) {
        arr[K] = i + min
        K++
        favor[i] = -
    }
}

```

SC: $O(\max - \min + 1)$

Google's Gmail : All Inboxes

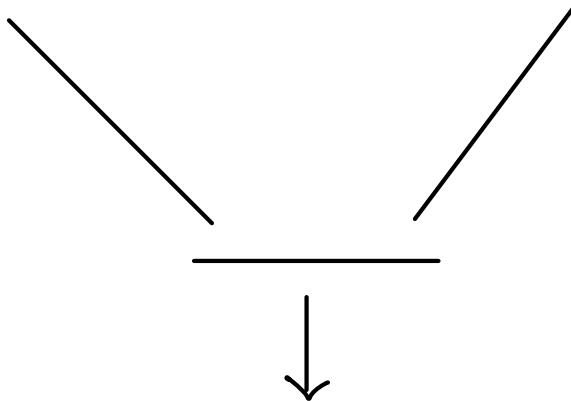
e1
e2

Sorted by time

← e1. 10am e1. 11am e1. 2pm e1. 4 pm

Sorted by time

← e2. 9am e2. 10:30 am e2. 4:30 pm



e2. 9am

e1. 10am

e2. 10:30 am

e1. 11am

:



Merge Two Sorted Arrays?

a[] → [2 4 7 8 12] → N *Sorted* $(-10^9 \leq \text{element} \leq 10^9)$

b[] → [3 5 6 7] → M *Sorted*

Merge
fun →

2	3	4	5	6	7	7	8	12
0	1	2	3	4	5	6	7	8



Idea - 1

- Form a C [] array : $N + M$ and fill it with all elements

2 4 7 8 12 3 5 6 7



Sort this array

TC : $(N + M) \log(N + M)$



Idea - 2

Ex 1:

$$a[] \rightarrow [\begin{matrix} \cancel{x} & \cancel{x} & \checkmark & \sim & \sim \\ 2 & 4 & 7 & 8 & 12 \end{matrix}] \dots 5$$

0 1 2 3 4

$$b[] \rightarrow [\begin{matrix} \cancel{x} & \cancel{x} & \cancel{x} & \cancel{x} & \checkmark \end{matrix}] \stackrel{=}{\bullet} b.length \dots 4$$

0 1 2 3

$$c[] = \frac{2}{0} \quad \frac{3}{1} \quad \frac{4}{2} \quad \frac{5}{3} \quad \frac{6}{4} \quad \frac{7}{5} \quad \frac{7}{6} \quad \frac{8}{7} \quad \frac{12}{8}$$

.

↑

Ex 2:

$$a[] = \frac{0}{1} \quad \frac{0}{5} \quad \frac{0}{6} \quad \frac{0}{9}$$

$$b[] = \frac{0}{2} \quad \frac{0}{4} \quad \frac{0}{8}$$

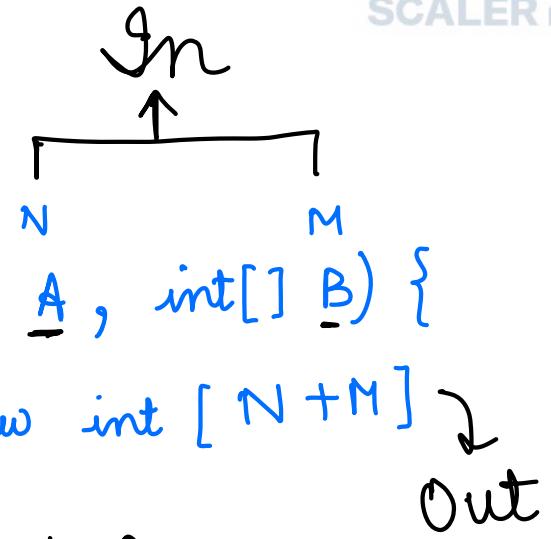
↑
end

$$c[] = \frac{1}{1} \quad \frac{2}{1} \quad \frac{4}{1} \quad \frac{5}{1} \quad \frac{6}{1} \quad \frac{8}{1} \quad \frac{9}{1}$$

↓



</> Code



int [] merge (int [] A, int [] B) {
 int [] C = new int [N+M] Out
 P1=0 P2=0 K=0
 while (P1 < N ll P2 < M) {
 if (A[P1] < B[P2]) {
 C[K] = A[P1]
 K++ P1++
 } else {
 C[K] = B[P2]
 K++ P2++
 }
 }
 while (P2 < M) {
 C[K] = B[P2]
 K++ P2++
 }
 while (P1 < N) {
 C[K] = A[P1]
 K++ P1++
 }
 return C

TC:

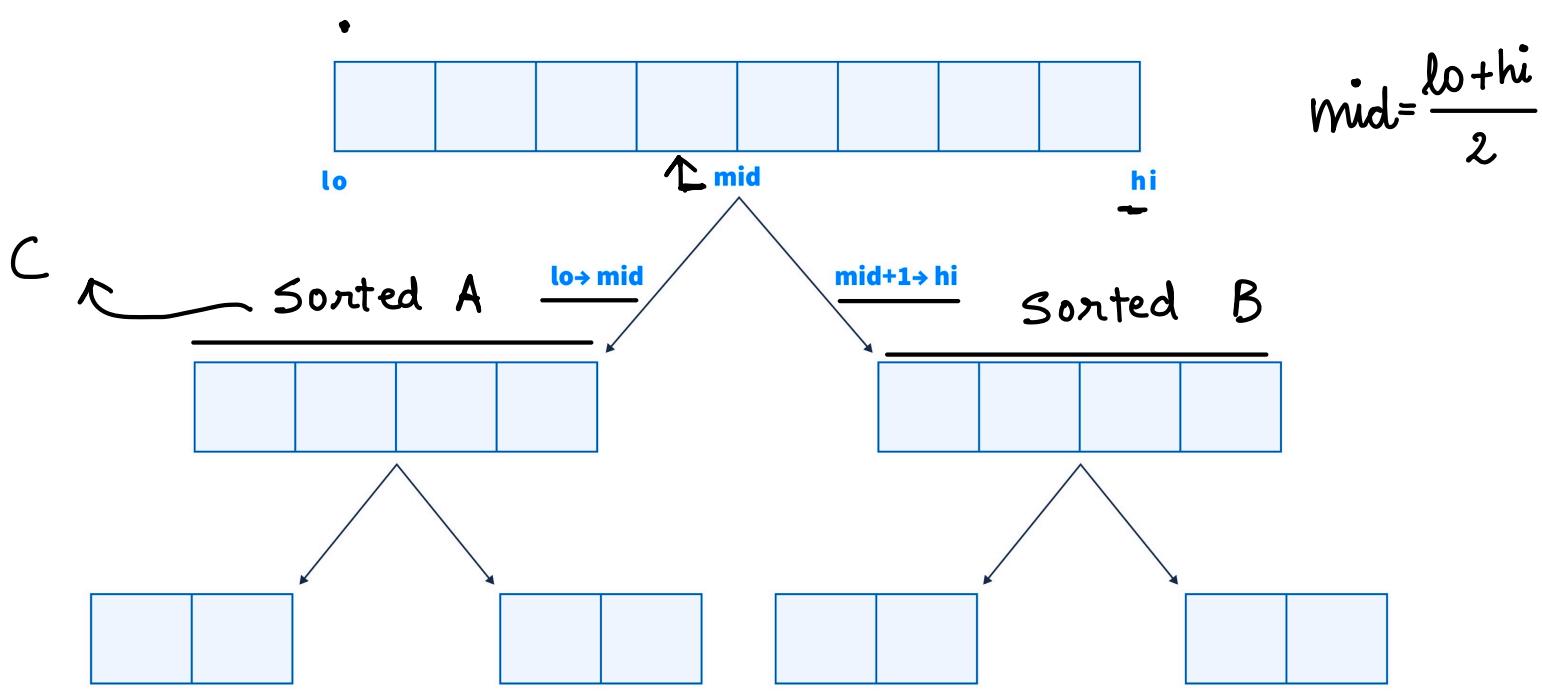
O(N+M)

SC:

O(1)



Merge Sort



$$A = \{ 3, 10, 6, 8, 15, 2, 12, 18, 17 \}$$

P = arr, 0, 8 and return sorted array
int []

$$\text{mid} = \frac{0+8}{2} = 4$$

SP1 arr, 0, 4 = A[] ↻ = {3, 6, 8, 10, 15}
SP2 arr, 5, 8 = B[] ↻ = {2, 12, 17, 18}

$$C \longrightarrow A + B$$

return C

$$A = \{ \frac{0}{3}, \frac{1}{10}, \frac{2}{6}, \frac{3}{8}, \frac{4}{15}, \frac{5}{2}, \frac{6}{12}, \frac{7}{18}, \frac{8}{17} \}$$

P = arr, 0, 8 and return sorted array
int []

$$\text{mid} = \frac{0+8}{2} = 4$$

$$\begin{array}{lll} \text{SP1} & \text{arr, } 0, 4 & = A[] \leftarrow \{ 3, 6, 8, 10, 15 \} \\ \text{SP2} & \text{arr, } 5, 8 & = B[] \leftarrow \{ 2, 12, 17, 18 \} \end{array}$$

$$C \rightarrow A + B$$

return C

int []

mergesort (int[] arr, int l, int h) {

$$m = \frac{l+h}{2}$$

int[] A = mergesort (arr, l, m);

int[] B = mergesort (arr, m+1, h);

int[] C = merge (A, B);

return C;

}



```

int []
1 → mergesort (int[] arr, int l, int h) {
    if (l == h) { int [] ba = new int [1]
        m =  $\frac{l+h}{2}$ 
        ba[0] = arr[l]
        return ba; }
2 → int[] A = mergesort (arr, l, m);
3 → int[] B = mergesort (arr, m+1, h);
4 → int[] C = merge (A, B);
5 → return C;
}
  
```

2, 3, 6, 8, 10, 12, 15, 17, 18

7	$l=2$	$h=2$
6	$l=1$	$h=1$
5	$l=0$	$h=0$
→ 4	$l=0$	$h=1$
	$m = \frac{0+1}{2} = \frac{1}{2}$	
3	$l=0$	$h=2$
.	$m = \frac{0+1}{2} = \frac{1}{2}$	$h=2$
2	$m=2$	
1	$m=4$	

1 arr →

3	10	6	8	15	2	12	18	17
0	1	2	3	4	5	6	7	8

 $\frac{1}{2}, 0, 8$
 $m=4$

A = 3 6 8 10 15

2 . l=0 h=4 m=2

A =

3	6	10
0	1	2

3

3	10	6
0	1	2

 m=1

A =

3	10
0	1

 B = 6

4

3	10	6
0	1	2

 m=0

5

3
0

10
1

 l=1 h=1

B =

2	12	17	18
5	6	7	8

2	12	18	17
5	6	7	8

8	15
3	4

8	15
3	4

2	12
5	6

18	17
7	8

8
3

15
4

2
5

12
6

18
7

17
8



int [] mergesort (int [] arr, int l, int h){
 if (l == h) { int [] ba = new int [1];
 ba[0] = arr[l];
 return ba; }
 m = $\frac{l+h}{2}$
 1 → int [] A = mergesort (arr, l, m);
 2 → int [] B = mergesort (arr, m+1, h);
 3 → int [] C = merge (A, B);
 4 → return C;
 5 → }

int [] mergesort (int [] arr, int l, int h){
 if (l == h) { int [] ba = new int [1];
 ba[0] = arr[l];
 return ba; }
 m = $\frac{l+h}{2}$
 3 → int [] A = mergesort (arr, l, m);
 int [] B = mergesort (arr, m+1, h);

int [] C = merge (A, B);

return C;

}

$$A = \{ 3, 10, 6, 8, 15, 2, 12, 18, 17 \}$$

another array → SORTED Output

New array

Extra Space

Same array

In Place Sorting

In next class:
→ Solve Merge Sort in inplace

Stable Sorting

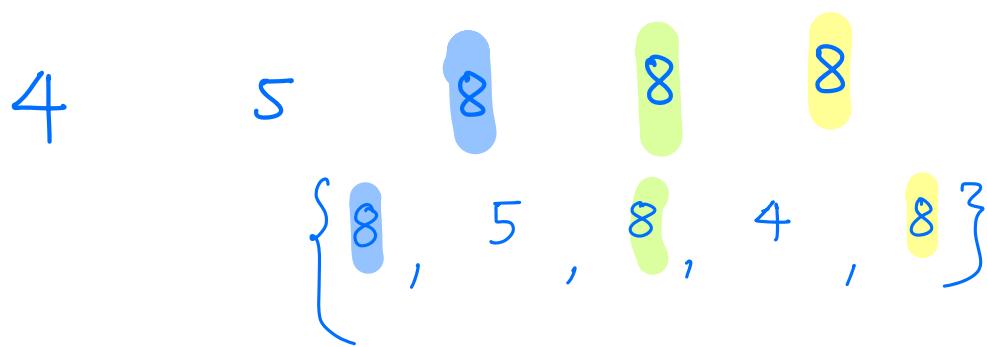
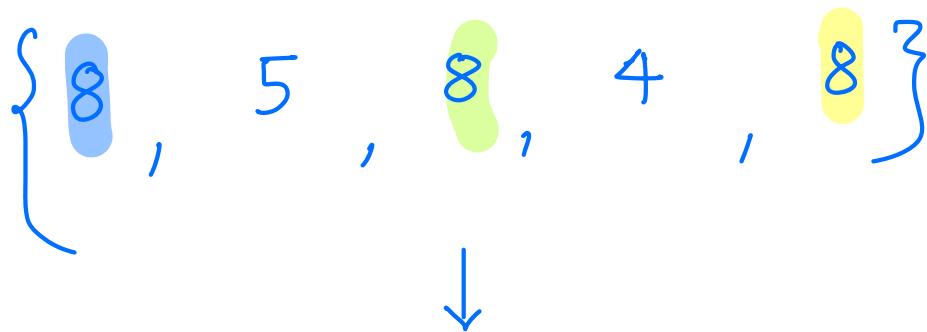
Relative order of equal elements doesn't change after sorting

$$\underline{A[]} = \{ 6, 5, 3, 5 \}$$

↓

$$\{ 3 \ 5 \ 5 \ 6 \}$$

Ex 2



HW

- bubble sort
- insertion sort
- Merge sort

[extra
space]