

* Radix Sort :

5934

$$4 \Rightarrow 5934 / 1 \% 10 =$$

$$3 \Rightarrow 5934 / 10 \% 10 =$$

$$9 \Rightarrow 5934 / 100 \% 10 =$$

$$5 \Rightarrow 5934 / 1000 \% 10 =$$

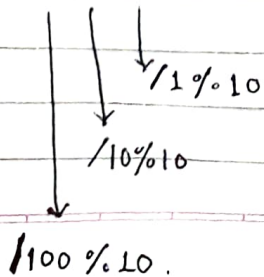
input arr :-

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

arr :-	3	2	9	7	2	0	7	2	0	3	2	9
1	4	5	7	3	5	5	3	2	9	3	5	5
2	6	5	7	4	3	6	4	3	6	4	3	6
3	8	3	9	4	5	7	8	3	9	4	5	7
4	4	3	6	6	5	7	3	5	5	6	5	7
5	7	2	0	3	2	9	4	5	7	7	2	0
6	3	5	5	8	3	9	6	5	7	8	3	9

exp $\Rightarrow 1$

exp = exp * 10



```

int exp = 1;
while (exp <= max)
    countsort(arr, exp)
    exp = exp * 10.
    
```

- Countsort (for $\text{exp} = 1$)

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

```

for (int i = 0; i < arr.length; i++)
    farr[arr[i] / exp % 10] ++
    
```

farr :-

0	0	1
1	0	
2	0	
3	0	
4	0	
5	0	1
6	0	1
7	0	2
8	0	
9	0	2

prefix sum of farr

farr:- 0	1	
1	0	
2	0	
3	0	
4	0	
5	2	1
6	3	2
7	5	4
8	0	
9	7	5

ans :- the length of ans' arr
is same as given arr length.

```

for (int i = arr.length; i >= 0; i--)
{
    pos = farr[arr[i] / exp % 10] - 1
    ans[pos] = arr[i];
    farr[arr[i] / exp % 10] --;
}
    
```

0	7 2 0	i = 5
1	3 5 5	i = 6
2	4 3 6	i = 4
3	4 5 7	i = 1
4	6 5 7	i = 2
5	3 2 9	i = 0
6	8 3 9	i = 3