RADIX SORT

- Radix Sort is dependent on Radix sort.
- Radix sort count sort ki stability property ko use krta hai!
- Count Sort me hum purey no. ko compare krty thy & Radix Sort me hum no. ke ek-ek ko compare krke sare no. ko sort krdety hai!

213	97	718	123	37	443

unsorted Array

↓ Radix Sort

37	97	123	213	443	718

Sorted Array

- Digits of a number, wheather it be (units digit, tens place, hundredth place), the digits will vary From 0 to 9:
  - Range = 10

HOW TO GET DIGITS ?

∴ Frequency array will be of Range 10 at maximum!

Comparing unit place

213
97
718
123
37
443
982
64
375
683

- Compare the unit digits & sort accordingly & if 2 no. have same unit digits, then arrange the no.s according to their stability

Comparing tens place

982
213
123
443
683
64
375
97
37
718

- Last one digit have been sorted
- Compare the tens place digits and if the tens place digit no. are equal, then arrange them accordingly to their stability

Comparing Hundredth place

213
718
123
037
443
064
375
982
683
097

- Last two digits have been sorted
- Compare the hundredth place digit of a no. and if there is no hundredth place digit then place digit then put 0 over there, if the hundredth place digit are same then compare stability of the no..

037
064
097
123
213
375
443
683
718
982

SORTED ARRAY

Digit at UNIT PLACE

* Divide the no. by (1) and get Quotient!
* Do modulus by (10) to get remainder!

7459 / 1 ⇒ 7459

7459 % 10 ⇒ (9)

Digit at TEN's PLACE

* Divide the no. by (10) and get Quotient!
* Do modulus by (10) to get the remainder!

7459 / 10 ⇒ 745

745 % 10 ⇒ (5)

Digit at HUNDREDTH's PLACE

* Divide the no. by (100) & get the Quotient!
* Do modulus by (10) to get the remainder!

7459 / 100 ⇒ 74

74 % 10 ⇒ (4)

UNIT PLACE se kaha tak LOOP chalega!? → SOLUTION

Sabse phele hum array me sabsay bada (Max.) no. nikal lengy! → KESE?

Max / Place Value > 0 → max = 4674 → 4674/1 %.10
4674/10 %.10
4674/100 %.10
4674/1000 %.10
4674/10000 %.10 < 0

Jab tak uper wali condition true tab tak LOOP chalega!

CODE - DRY RUN

```
public static void radix sort (int [] arr){
  int max = arr[0];
  for(int i=1; i<arr.length; i++) {
      if(arr[i] > max) {
          max = arr[i];
      }
  }
  int place = 1; → Aabhi unit place hai
  while(max/place > 0)
  {
    CountSort(arr, place);
    place = place * 10;
  }
}
public static void countSort
          (int [] arr, int place) {
int farr[] = new int[10];
int ans [] = new int [arr.length];
for(int i=0; i<arr.length; i++){
  int value = arr[i] / place %.10;
  farr [value] ++;
}
for(int i=1; i<farr.length; i++) {
   farr[i] = farr[i] + farr[i-1];
}
```

Hume max. no. mil gaya

7449
2265
2786
3232
8529
9383
2174

(n)

phele one's place ke liye COUNT SORT krengy, fir place *10
Aab ten's place ke liye COUNT SORT krengy, fir place * 100
Aab Hundredth's place ke liye Count sort krengy,
Aur yeh loop tab tak chalega tab tak max / place > 0 .

(n)

we need to collect the frequencies!

(n)

now, we need to convert the frequency array to the prefix-sum array or the cumulative frequency array!

Aab humara farr → prefix-sum array ban gaya hai!
Now, we need to fill the ans array!

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

for(int i=0; i<ans.length; i++){
    arr[i] = ans[i];
}
System.out.print("After sorting on" + place + "place →");
print(arr);
}

public static void print(int[] arr){
  for(int i=0; i<arr.length; i++){
      System.out.print(arr[i] + " ");
  }
  System.out.println();
}

public static void main(String[] args){
Scanner s = new Scanner(System.in);
int n = s.nextInt();
int[] arr = new int[n];
for(int i=0; i<n; i++){
    arr[i] = s.nextInt();
 }
radixsort(arr);
print(arr);
}
}
```

(n)

(n)

farr.length
↑
range

arr.length

Time Complexity = digits * (n + r)

∴ T(n) of Radix Sort ≃ d(n) ⇒ 9 * 30

∴ let d = 9
if n = 30

T(n) of Merge Sort = $n \log n = 30 \log_2 2^{30}$

∴ T(n) of Radix Sort is much² better!

T(n) of Radix Sort <<< T(n) of Merge Sort