Q→ Find the smallest number that can be formed by rearranging the digits of the given input. A[i] → digit [0 9]

A = [ 1 3 5 2 3 ]  (0 1 2 3 4)

O/P → 1 2 3 3 5

sort in ascending order.

TC = O(N log (N))

A = [ 1 5 2 1 3 0 5 1 ]  (0 1 2 3 4 5 6 7)

O/P → 0 1 1 1 2 3 5 5

0...0 1...1 2... 3.. 4... 9...9 ✓

F[i] → freq of $i^{th}$ element

A = [ 1 3 8 3 2 6 5 3 8 ]  (0 1 2 3 4 5 6 7 8)

F = [ 0 1 1 3 0 1 1 0 2 0 ]  (0 1 2 3 4 5 6 7 8 9) ✓ ← 9

```
for i → 0 to (N-1)
    F[A[i]] += 1
```

TC = O(N)

SC = O(Range) = O(10) = O(1)

```
for i → 0 to 9 ←
    for j → 1 to F[i] }
        print(i)
```

TC = O(Range + N) = O(N)

SC = O(1)

$\sum_{i=0}^{9} F[i] = N$

Total → TC = O(N) SC = O(1)

Count Sort

---

If A[i] <= $10^9$ → X → use merge/quick sort.

A = [ ... 11 ... 328765 ... $10^9$ ... ]

If $-10 <= A[i] <= 10 \rightarrow$

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| A = [ | 3 | -8 | -10 | 3 | 0 | 10 | 2 | -6 | -8 ] |

Range [-10 10] → 21

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | ... | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F = [ | 1 | 0 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 2 | 0 | ... | 1 ] |
| | -10 | -9 | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | | 10 |

F[i] → freq of i-10 (i + min) (i + (-10))

i → i-10

i+10 → i ✓

```
for i → 0 to (N-1)
    F[A[i] + 10] += 1
```

TC = O(N)

SC = O(Range) = O(21) = O(1)

```
for i → -10 to 10
    for j → 1 to F[i + 10]
        print (i)
```

TC = O(N)

SC = O(1)

Total → TC = O(N) SC = O(1)

---

N = 368

2 1 0 →

N % 10

(N/10) % 10

(N/100) % 10

$i^{th} \rightarrow (N/10^i) \% 10$

Inbuild sorting functions

Comparator

```
int compare (A, B) {
    return -ve number → A < B
    return 0 → A = B
    return +ve number → A > B
}
```

Q → Sort the integer array wrt $K^{th}$ digit of the number.

A = [ 326, 18, 523 ] K = 0

O/P → 523 326 18

Sorting → TC = O(N log (N))

A = [362, 399, 318] K=2

O/P → 362, 399, 318

equal

A = [361, 432, 12, 78, 500, 112] K=1

F[i] → list of elements where $K^{th}$ digit is i

- F[0] → {500}
- F[1] → {12, 112} ordered
- F[2] → {}
- F[3] → {432}
- F[4] → {}
- F[5] → {}
- F[6] → {361}
- F[7] → {78}
- F[8] → {}
- F[9] → {}

Σ length of lists = N

TC = O(N)

SC = O(N)

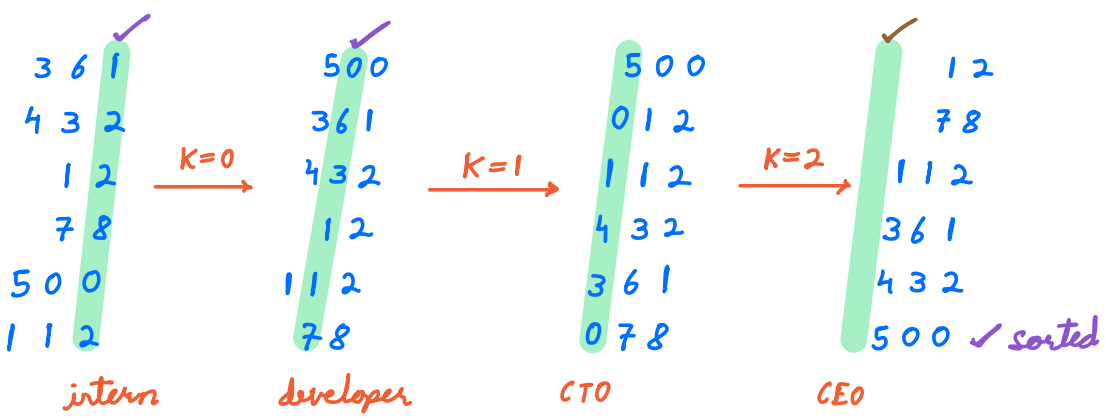
500 12 112 432 361 78

---

**Radix Sort** → sort digit by digit

38 < 42

A = [361, 432, 12, 78, 500, 112]




TC = O(N * #digits) SC = O(N)

$N = 10^5$ $A[i] <= 10^9$

$N * \#digits = 10^5 * 10 = 10^6$

$N * \log_2(N) = 10^5 * \log_2(10^5) = 10^5 * 17 = 1.7 * 10^6$

9 8 7 3 1 2 5 9 6 → 9

1 0 0 0 0 0 0 0 0 0 → 10

$2^x = 10^5$

Inbuild functions do not use radix sort →

do not compare elements ⇒ no custom comparator.

data type → double

3112.368256 → ✓

3112.368255 → ✗

---

10:50 PM

Q→ Find the sum of (max − min) ∀ subsets of the array.

continuous & non-continuous

A = [3 2 5] (0 1 2) → 2*2*2 = $2^3$

Total # subsets = $2^N$ ✓

| | | |
|---|---|---|
| [3] | 3 − 3 = 0 | |
| [2] | 2 − 2 = 0 | 3*(2−2) = 0 |
| [5] | 5 − 5 = 0 | 2*(1−4) = −6 |
| [3 2] | 3 − 2 = 1 | 5*(4−1) = 15 |
| [3 5] | 5 − 3 = 2 | 9 ✓ |
| [2 5] | 5 − 2 = 3 | |
| [3 2 5] | 5 − 2 = 3 | |

Contribution Technique

Ans = $\sum_i$ contribution of A[i]

9 (Ans)

$$Ans = \sum_i A[i] * (\#\text{subsets where } A[i] \text{ is max} - \#\text{subsets where } A[i] \text{ is min})$$

---

A = [3 2 8 7 4 6] (0 1 2 3 4 5)

# subsets where 6 is max = $2^3$ = 8 ✓ → 2 (#elements < A[i])

A = [3 2 8 7 4 6] (0 1 2 3 4 5)

# subsets where 6 is min = $2^2$ = 4 ✓ → 2 (#elements > A[i])

sort

A = [2 3 4 | 6 | 7 8] (0 1 2 | 3 | 4 5)

$6 * (2^i - 2^{N-1-i})$

$$Ans = \sum_{i} A[i] * (2^{i} - 2^{N-1-i})$$ → after sorting A

TC = $O(N \log(N))$

```
ans = 0
x = 2^0        y = 2^(N-1)
for i → 0 to (N-1)
    ans += A[i] * (x - y)
    x *= 2
    y /= 2
```

H.W → Try this if array has duplicates.