arr → 40 30 2 10 60

10   20   30   40   50

i=1   j=i — ①

i=2   j=2 — ②

i=3   j=3 — ②

i=4   j=4 — ①

Ô(n)

Best case
↓
array is
sorted

Bubble Sort] → O(n²)
Selection Sort]

Insertion Sort] → O(n²) — worst/Avg.
                    O(n) → Best
                    Sorted array.

Optimised

Inplace chaye

arr[j+1] ≤ arr[j]

stopping

j=1

① 0.0   30 40 20 10 35

30 40   20 10   35

② 0.0   30 40 20 10 35

0.1   30 20 40 10 35

20 30 40 10 35

③ 3.0   10 20 30 35 40 j'
order

3.1   10   20 30 35 40

④ 2.0   20 30 40 10 35

2.1   20 10 30 40 35

2.2   10 20 30 40 35

10 20 30 40 35

$hi \to lo$ $0 \leq arr[i] \leq hi$ Size of array $\leq 10^5$

$arr \to \{ 0, \textcircled{1}, \textcircled{5}, 5, 3, \textcircled{9}, 8, 6, 7, 4, 7, 9, \overset{hi}{\textcircled{10}}, \textcircled{10}, 9, 6, 4, 3, 3, 2, 1, 0, 7, 6, 5 \}$

Sort $\to$

$\{ 0, 0, 1, 1, 2, 3, 3, 3, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 7, 8, 9, 9, 9, 10, 10 \}$

freq. Map $\to$

$\textcircled{0} \to \textcircled{2}$ freq val.

Size = hi+1

$\textcircled{1} \to 2$

$2 \to 1$

$3 \to 3$

$4 \to 2$

$5 \to 3$

$6 \to 3$

$7 \to 3$

$8 \to 1$

$9 \to 3$

$10 \to 2$

Range $0 < 10^6$

complexity -

$\longrightarrow$ fmap $\to$ $O(n)$

$\longrightarrow$ fill $\to$ $O(n)$

| overall - $O(n)$ |

Range $<<<$ Size of array

Step ①   Generate freq. map using array.

Step ②   fill the array.

count Sort — lo, hi

$-2 \quad 10 \quad \longrightarrow$ arr = { 10, 7, 6, 5, -1, -2, ③, 0, 4, 9, 8, 6, 6, -1, -2, 7, 10, 10, 4}

-2, -2, -1, -1, 0, 3, 4, 4, 5, 6, 6, 6, 7, 7, 8, 9, 10, 10, 10

fmap = new int [hi - low + 1] $\longrightarrow$

$= 10 - (-2) + 1 = 13$ $\longrightarrow$ Sorted

doesn't Exist

gndx	val	freq
fmap'		
0 →	-2 →	2
1 →	-1 →	2
2 →	0 →	1
3 →	1 →	0
4 →	2 →	0
5 →	3 →	1
6 →	4 →	2
7 →	5 →	1
8 →	6 →	3
9 →	7 →	2
10 →	8 →	1
11 →	9 →	1
12 →	10 →	3

How to get index to
fill fmap'

index = arr[i] - min

fmap[index] ++;

How to get value from
fmap.

val = i + min

$1 + (-2) = \boxed{-1}$

$0 \rightarrow 9$

$$gp \begin{cases} arr \rightarrow \{ 7 \quad 6, 3, 2, 1, 0, 4, 5, 6, 7, 9, 4, 3, 2, 1, 2, 0, 6, 9, 8, 3, 8, 9, 6, 7\} \\ char \rightarrow \{ a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y \} \end{cases}$$

0 — a
1 — b
2 — c
3 — d
4 — e
5 — f
6 — g
7 — h
8 — i
9 — j

op → arr → 0 0 1 1 2 2 2 3 3 3 4 4 4 5 6 6 6 6 7 7 7 8 8 9 9

f j e o d n p c m u g l w h b i r x a j y t v k s

Stability → Maintain order, if element occur more than one

① What?
② How?
③ Why?

Index → 0 1 2 3 4 5 6 7 8 9 10 11

arr → (3), (7), (6), (2), (8), (9), (4), (5), (7), (6), (5), (3)

char → a b c d e f g h i (j) (k) l

| n arr → | 2 | 3 | 3 | 4 | 5 | 5 | 6 | 6 | 7 | 7 | 8 | 9 |
| n chan → | d | a | l | g | h | k | c | j | b | i | e | f |

Total no. of

index	Val	freq →	Prefix sum array
0	2 →	0 → 1	0 −1
1	3 →	2 → 2	1 0
2	4 →	1 → 3 4	2
3	5 → 2 →	5 6	5 4 3
4	6 → 2 →	8	7 6 5
5	7 → 2 →	10	8 7
c	8 → 1 →	11	10 9
7	9 → 1 →	12	11 10

Because it is
Group ①

Stability ??

Step ④
Fill original array

mine val mall

Step ①
generate fmap

Step ②
create a
prefix sum
using fmap

Step ③
travel from
end on one
place that
& Reduce fmap (psum)

① for(int i=1; i < n, i++)

② for(int i=0; i*i < n; i++)

③ for(int i=1; i < n; i *= 2)

④ for(int i=n; i > 1; i = i/2)

⑤ for(int i=1; i <= n; i += m)
        for(int j=1; j < m; j++)

⑥ j=1
   for(int i=1; i <= n;) {
         if(j == i) {
              j=1;
              i++;
         } j++;
   }

⑦ int i=1;
   s=1;
   while(s < n) {
        i++;
        s= s+i;
   }