1	2	7	3	4	7	9	8	6	5	9	1
a	b	c	d	e	f	g	h	i	j	k	l
0	1	2	3	4	5	6	7	8	9	10	11

Sort →

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

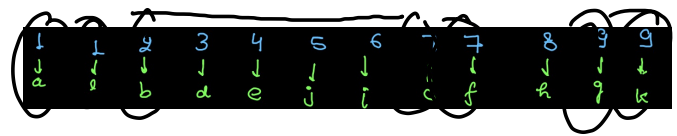
min = 1, max → 9

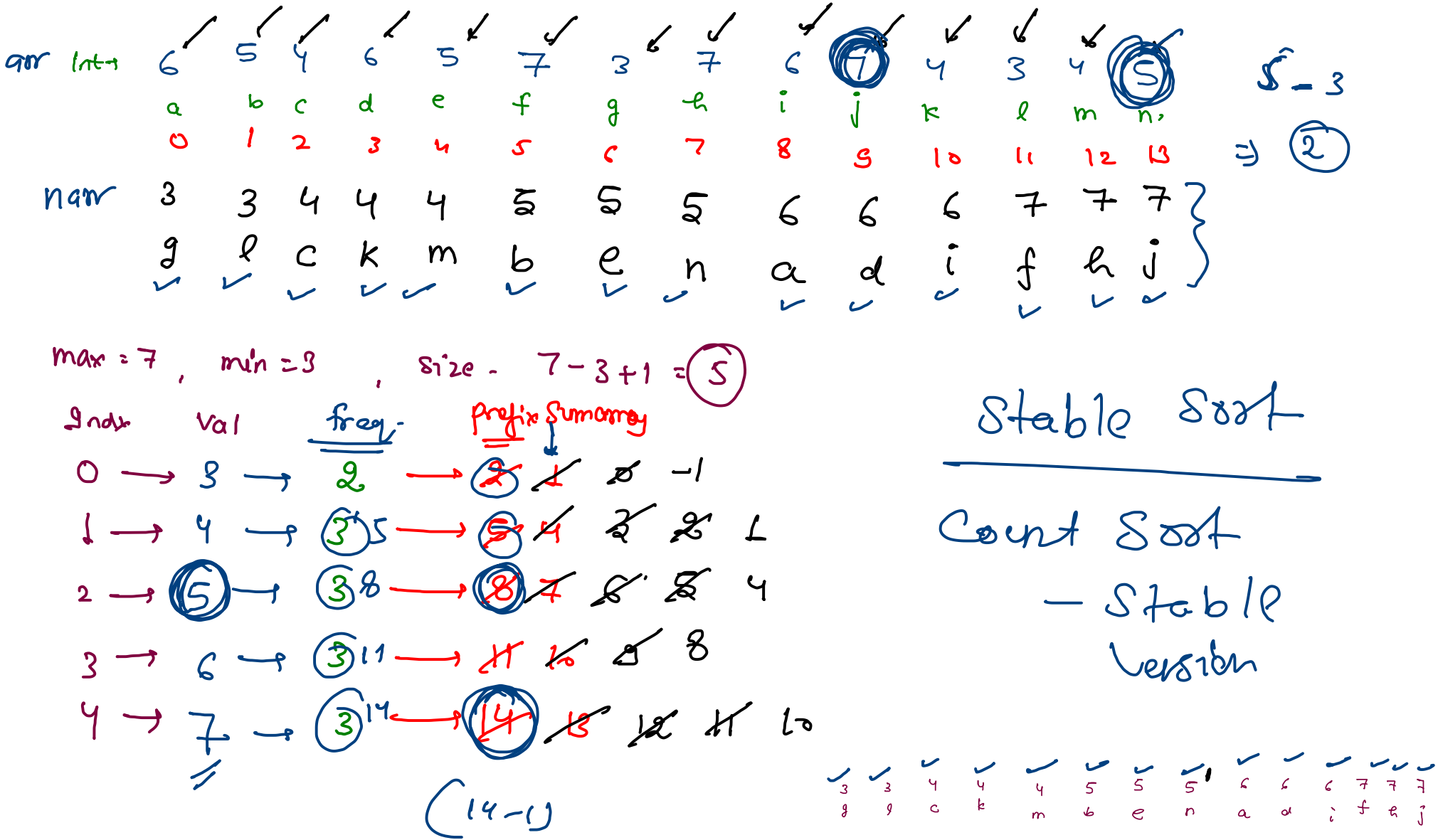
Size = 9 - 1 + 1 = 9

fmap →

Index	Value	freq.	prefix sum array
0	1	2	0
1	2	1	2
2	3	1	3
3	4	1	4
4	5	1	4
5	6	1	5
6	7	2	7
7	8	1	8
8	9	2	9

Stability →




arr int→ 6 5 4 6 5 7 3 7 6 7 4 3 4 5
a b c d e f g h i j k l m n
0 1 2 3 4 5 6 7 8 9 10 11 12 13
S-3 ⇒ 2
narr 3 3 4 4 4 5 5 5 6 6 6 7 7 7
g l c k m b e n a d i f h j
max = 7, min = 3, size - 7-3+1 = 5
Indx Val freq Prefix Summary
0 → 3 → 2 → 2 1 0 -1
1 → 4 → 3 5 → 5 4 2 2 1
2 → 5 → 3 8 → 8 7 6 5 4
3 → 6 → 3 11 → 11 10 9 8
4 → 7 → 3 14 → 14 13 12 11 10
(14-1)
Stable Sort
Count Sort
- Stable
Version
3 3 4 4 4 5 5 5 6 6 6 7 7 7
g l c k m b e n a d i f h j

Radix Sort

0 - 9

3 6 9 2

4 8 5 3

9 6 4 ← 1st

2 7 3 0 → Count

8 9 6 2

5 4 1 7

4 3 2 8

1 6 0 4

9 7

Sort on the basis of ith digit & maintain order

2 7 3 0

3 6 9 2

8 9 6 2

4 8 5 3 ← 2nd

9 6 4

1 6 0 4

5 4 1 7

9 7

4 3 2 8

1 6 0 4

5 4 1 7

4 3 2 8

2 7 3 0 ← 3rd digit

4 8 5 3

8 9 6 2

0 9 6 4

3 6 9 2

0 0 9 7

$d \times n$

$O(n)$

0 0 9 7

4 3 2 8

5 4 1 7

1 6 0 4

3 6 9 2 ← 4th digit

2 7 3 0

4 8 5 3

8 9 6 2

0 9 6 4

9 7

9 6 4

1 6 0 4

2 7 3 0

3 6 9 2

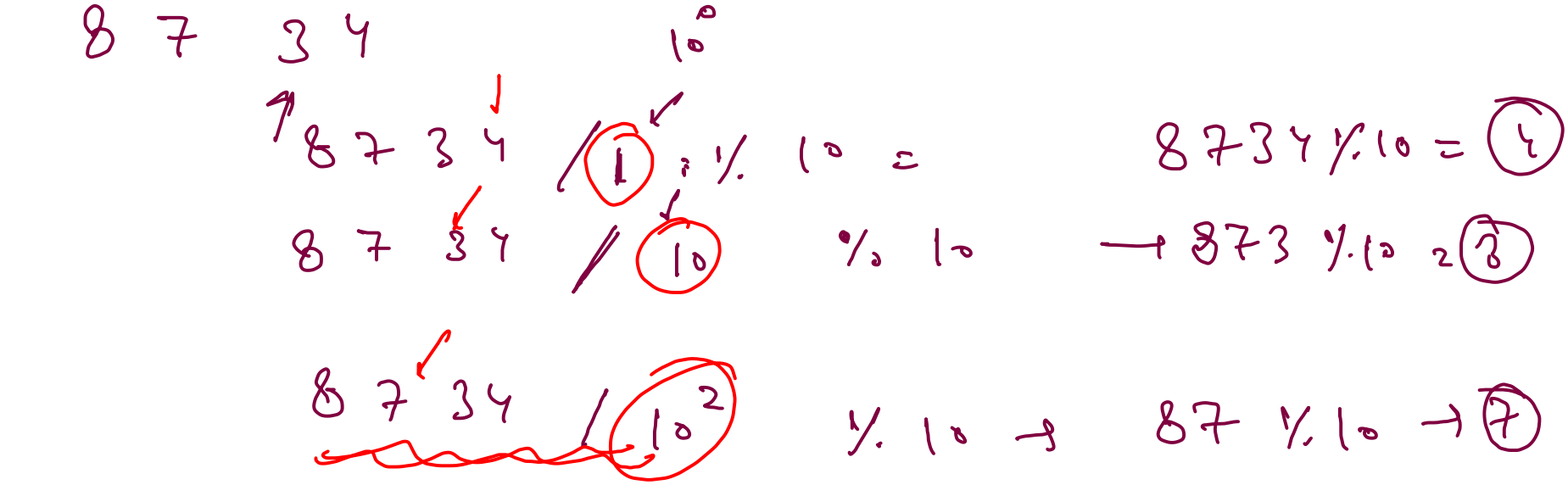
4 3 2 8

4 8 5 3

5 4 1 7

8 9 6 2

8 7 3 4



Num / Exp % 10 → digit Count ++

Target Sum Pair

target = 70

90 50 20 10 30 40 60 70 80

n log n + n → O(n log n)

for 1/0

Sort → 10 20 30 40 50 60 70 80 90

i < j] Run

i ... j

i > j → stop

{
arr[i] + arr[j] == target → Print pair, i++; j--;

arr[i] + arr[j] > target → j--; Sum Reduced

arr[i] + arr[j] < target → i++ Sum Increased
}

10 60
20 50
30 40

10 90
20 80
30 70
40 60

2 - 90

7-11

Loops.

1. Complexity Analysis
2. Loop Discussion
3. Pivot Rotated
4. Segmented Sieve

(1)
```
for (int i = 1; i <= n; i++)
```

(2)
```
for (int i = 1; i*i <= n; i++)
```

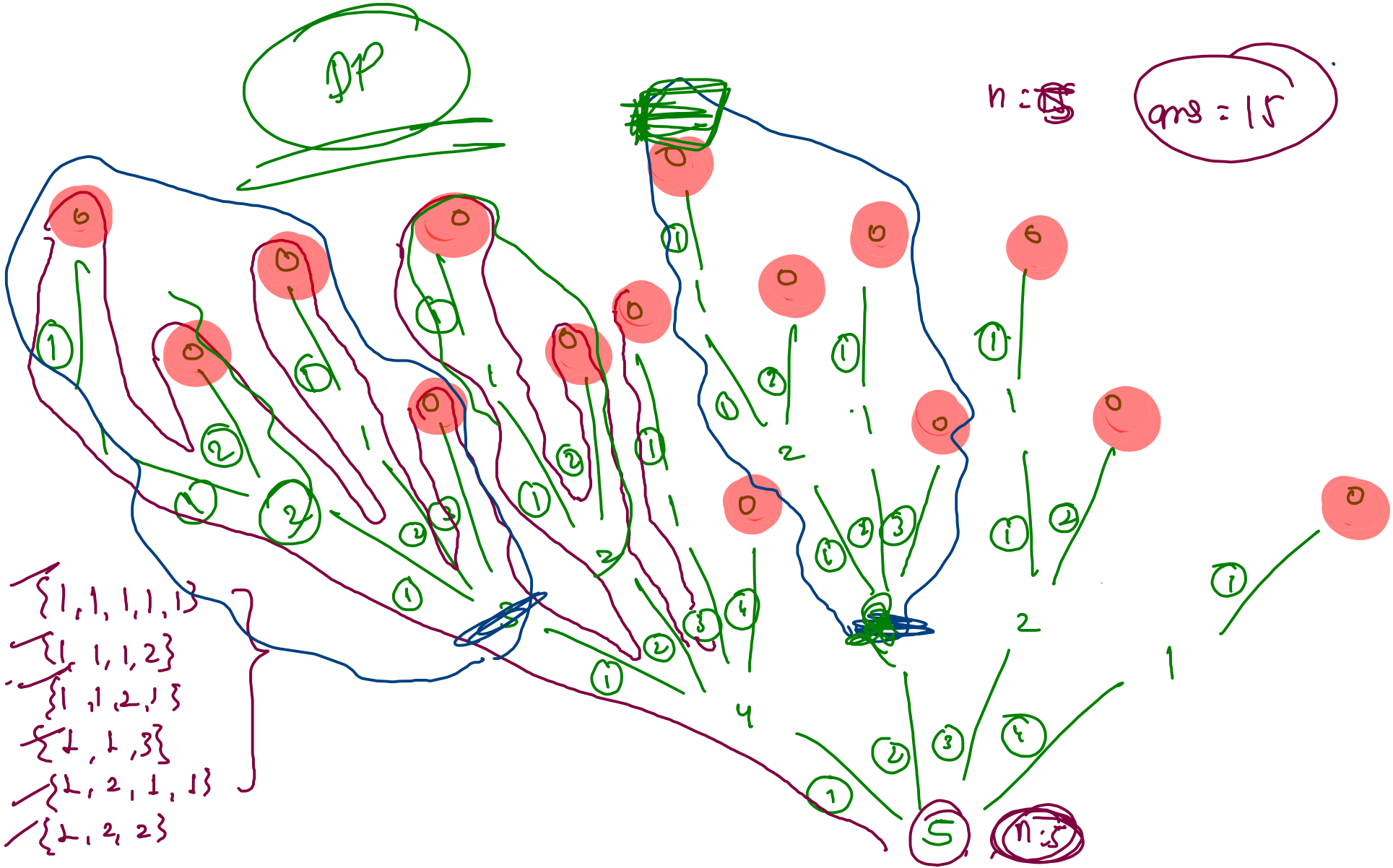
(3)
```
for (int i = 1; i <= n; i *= 2)
```

(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;
}
```


DP
n = 5
ans = 15
{1, 1, 1, 1, 1}
{1, 1, 1, 2}
{1, 1, 2, 1}
{1, 1, 3}
{1, 2, 1, 1}
{1, 2, 2}
5
n:5