Maximum Swap

Sunday, 5 September 2021 6:42 PM

Number → 9 9 8 8 7 5 6 7 3 8 4

8 7

max. possible no. with single swapping.

if swapping is possible?

→ 9 8 7 6 5 4

next greater - 11

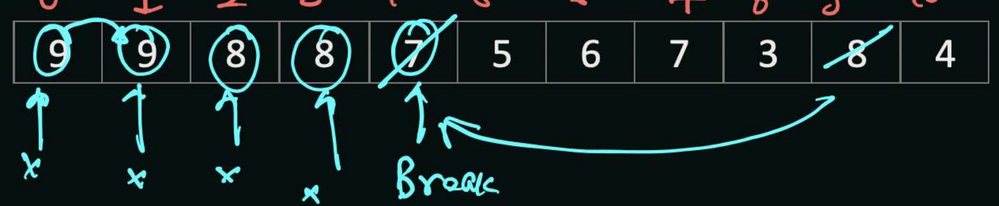
How to ensure?

MSD → Max. No. with single swapping.

→ 9 9 8 8 8 5 6 7 3 7 4

Approach - (1)

number in character array -



Right Greatest index →

1 1 9 9 9 9 9 9 9 9 10

Max. with swapping is possible when swapping takes place in MSD side

number → 9 9 8 8 8 5 6 7 3 7 4

Time → O(n)

space → O(n)

⇒ constant space.

Approach-2.

Time → O(n)

Space → constant space

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

assumption arr[i] = 5 arr[5] = 5

i = 5

→ valid digits for swapping - 9, 8, 7, 6

9 index is greater than i

→ last occurrence of decreasing priority order.

arr[5]

last occurrence index of digit → array of size = 10

0 → -1

1 → -1

2 → -1

3 → 8

4 → 10

5 → 5

6 → 6

7 → ~~4~~ 7

8 → ~~2~~ ~~3~~ 9

9 → ~~8~~ 1

① Make an array of size 10 of digits

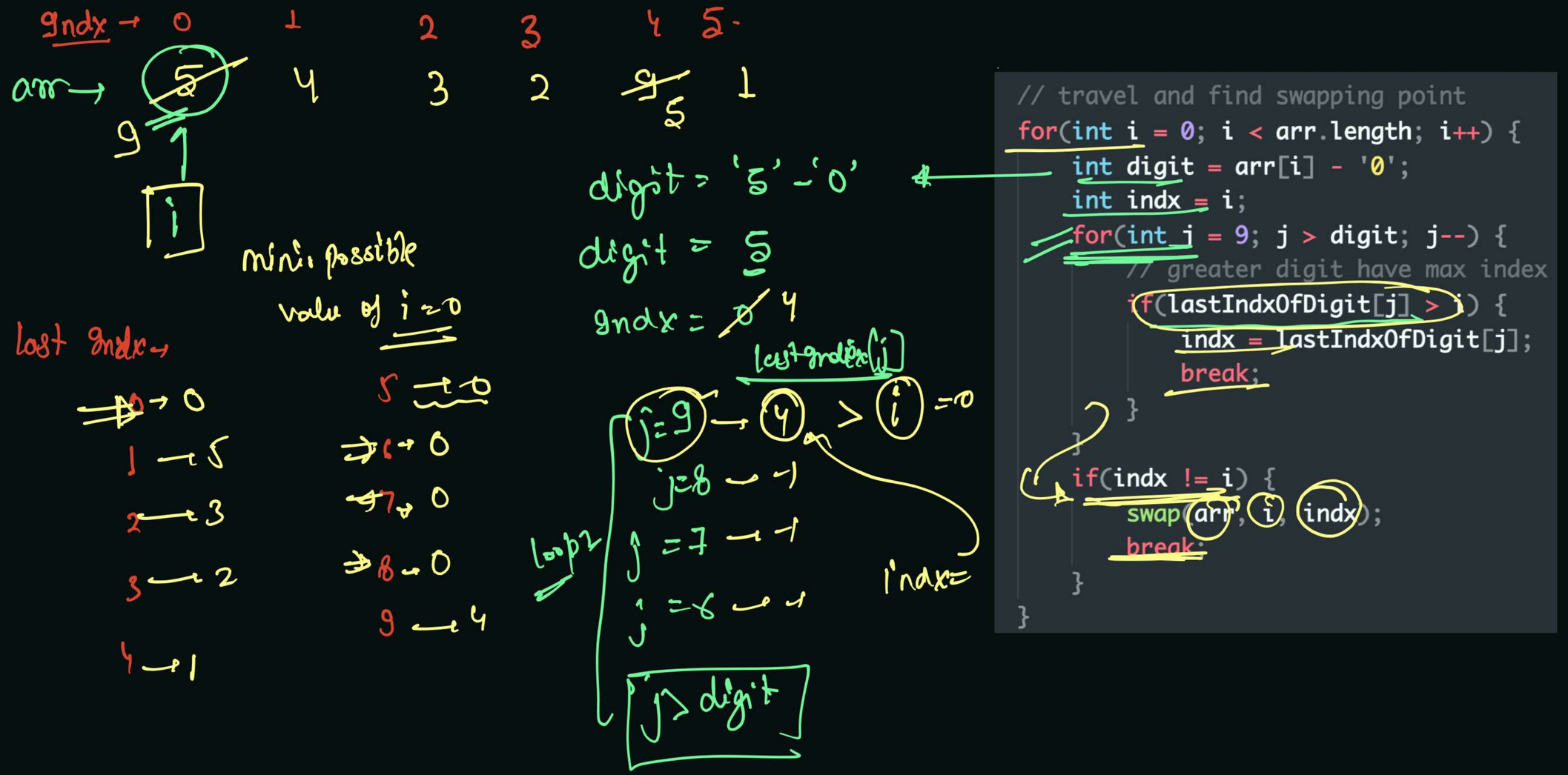
② Travel in array and fill last occurrence digit's array.

③ Make one more iteration and check swapping point.

→ space → 10 size

→ time → 9 n worst case

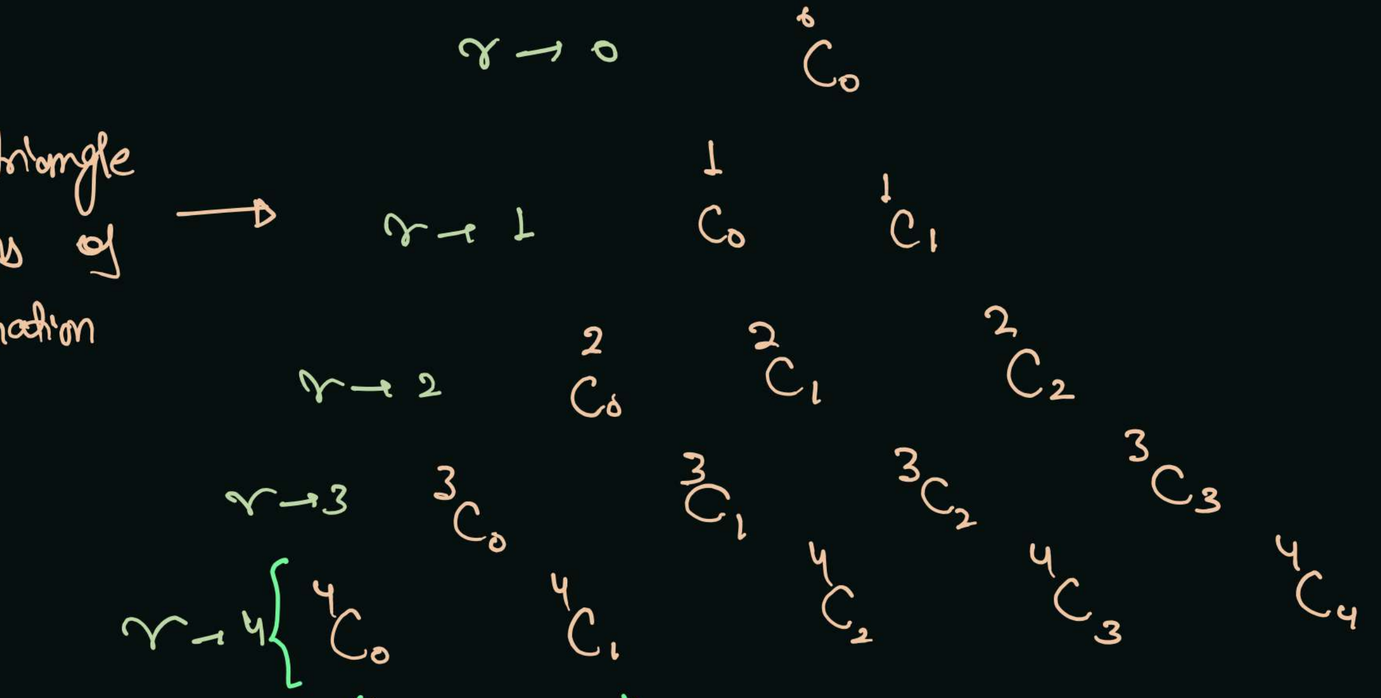
O(n)




```java
// travel and find swapping point
for(int i = 0; i < arr.length; i++) {
    int digit = arr[i] - '0';
    int indx = i;
    for(int j = 9; j > digit; j--) {
        // greater digit have max index
        if(lastIndxOfDigit[j] > i) {
            indx = lastIndxOfDigit[j];
            break;
        }
    }
    if(indx != i) {
        swap(arr, i, indx);
        break;
    }
}
```

Pascal's Triangle 2

Sunday, 5 September 2021 7:30 PM



Given a row, return all elements in it in List<Integer>

prev Curr

$curr = prev * factor$

$^nC_r * factor = {}^nC_{r+1}$

$prev * factor = curr$

Factor = ?

$^{n}C_{r}$ and $^{n}C_{r+1}$ have a relation,

How to find relation factor

$^{n}C_{r} * \text{Factor} = {}^{n}C_{r+1}$

Permutation and Combination

$a! = a \times a-1 \times a-2 \times \cdots \times 1$

$\frac{n!}{(n-r)!\,r!} \times F = \frac{n!}{(n-r-1)!\,(r+1)!}$

$F = \frac{(n-r) \times (n-r-1)! \times r!}{(n-r-1)! \times (r+1) \times r!}$

$$F = \frac{n-r}{r+1}$$

$^{n}C_{0} = 1$ always

$\frac{1}{\infty} = 0.0\ldots1 \approx 0$

$\frac{1}{\infty} = 0$

$\frac{1}{0} = \infty$

$^{n}C_{0} = \frac{n!}{n!\,0!}$

$0! = 1$ · we can prove that

$= \frac{1}{0!}$

$\frac{1}{0}\infty = \frac{1}{1} = 1$

$\frac{1}{\infty} = 0$

$^{n}C_{n} = \frac{n!}{0!\,n!} = 1$

Curr * Factor = next

Property $* \; {}^{n}C_{r} \times \frac{n-r}{r+1} = {}^{n}C_{r+1}$

$* \; {}^{n}C_{0} = 1 \quad * \; {}^{n}C_{n} = 1 \quad * \; {}^{n}C_{r} = {}^{n}C_{n-r}$

Factor = $\frac{n-r}{r+1}$

Row = 3 $\quad {}^3C_0 \quad {}^3C_1 \quad {}^3C_2 \quad {}^3C_3$ $\quad$ 1, 3, 3, 1

n = 3

r = 0 | val = 1 → push in result and change val , val = $1 * \frac{3-0}{0+1}$

r = 1 | val = 3 → " , val = $3 * \frac{3-1}{1+1} = 3$

r = 2 | val = 3 → " val = $3 * \frac{3-2}{2+1} = 1$

r = 3 | val = 1 → " val = $1 * \frac{3-3}{3+1} = 0$

val = 0

Result → 1 3 3 1

$$n! = \frac{(n+1)!}{(n+1)}$$

put value of n as '0'

$$0! = \frac{(0+1)!}{0+1} = \frac{1!}{1!}$$

$$\boxed{0! = 1}$$

$$\boxed{a^0 = 1}$$

$$\boxed{a^{-1} \cdot a^1} = a^{-1+1} = a^0$$

$$\frac{a^1}{a^1} = 1 \qquad \boxed{= a^0}$$

Complex Number Multiplication

Sunday, 5 September 2021 8:15 PM

Complex Number → real + imaginary 'i'

Ex → ① 5+4i
② -5+3i
③ 4+-5i
④ -3+-2i

find a_1, a_2, b_1, b_2

Result

Real + imaginary i

substring a_1 → 0 to index of 'i'

b_1 → index of 'i' +1 to length

complex Num1 = $a_1 + b_1 i$

complex Num2 = $a_2 + b_2 i$

Similarly a_2 b_2

result = num1 * num2

$= (a_1 + b_1 i) * (a_2 + b_2 i)$

$= a_1 * a_2 + a_1 * b_2 i + a_2 * b_1 i + b_1 * b_2 i^2$

$= (a_1 - a_2 - b_1 - b_2) + (a_1 - b_2 + a_2 - b_1)\, i$

Real | imaginary

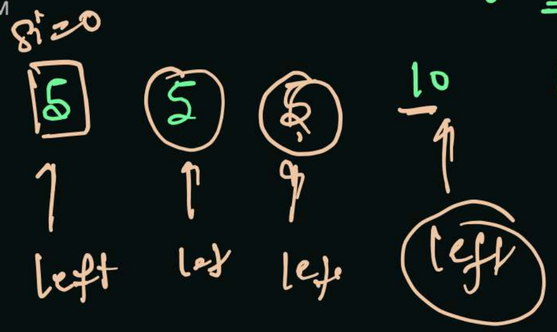
Proof:

$i^2 = -1$

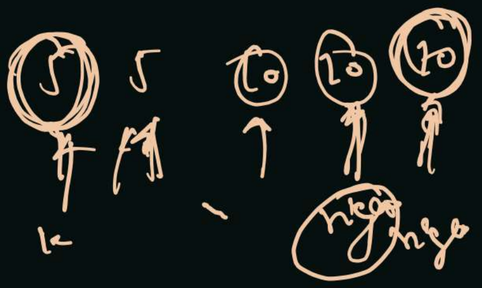
2 sum : Target Sum Unique Pair

Sunday, 5 September 2021 8:42 PM

target = 60



10 10 20 30 30 40 50 50



[preprocessing]

```
sum = arr[left] + arr[right]
if(sum == target){
    pair found
    left++; right--;
}else if(sum > target){
    right--;
}else{
    left++;
}
```

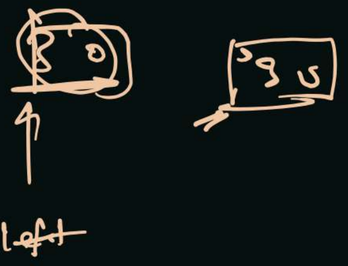
Same left pt right--

Smartly

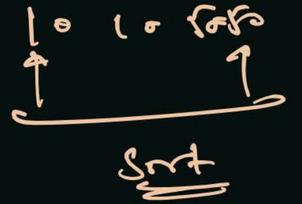
Pre processing.

```
if( left != 0 && arr[left] == arr[left-1])
    left++;
    continue;
}
```

10 50 50 10



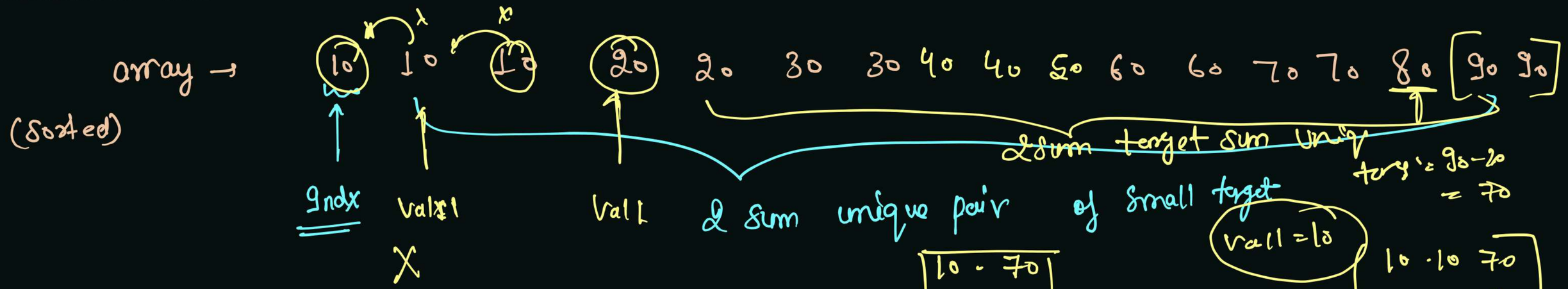
30 30 30



3 sum : Target Sum Unique Triplet

Sunday, 5 September 2021 8:42 PM

target = 90 Triplet , Time - $O(n^2)$ Brute force $O(n^3)$



2 sum unique pair of small target

10 - 70
20 - 60
30 - 50
40 - 40

add val1 in all list →

val1 = 10

10	10	70
10	20	60
10	30	50
10	40	40

Triplet → val1 + val2 + val3 ≡ target

val1 = arr[indx]

Small-target = target - val1;

val1 = 10

Small target = 80

val1 = 20

② + triplet ③ 5-2

0 1 2 3 4

Comparation b/w triplets and pair → freeze val1 and find pair from rest of array for triplet