tcs | NQT

NATIONAL QUALIFIER TEST

PROGRAMMING LECTURE 3

CS

Imp Announcements :

1) Tomorrow ⟶ Apti session (7 PM)

2) TCS NQT Admit Card is out.

About Aditya Jain sir

1. Appeared for GATE during BTech and secured AIR 60 in GATE in very first attempt - City topper
2. Represented college as the first Google DSC Ambassador.
3. The only student from the batch to secure an internship at Amazon. (9+ CGPA)
4. Had offer from IIT Bombay and IISc Bangalore to join the Masters program
5. Joined IIT Bombay for my 2 year Masters program, specialization in Data Science
6. Published multiple research papers in well known conferences along with the team
7. Received the prestigious excellence in Research award from IIT Bombay for my Masters thesis
8. Completed my Masters with an overall GPA of 9.36/10
9. Joined Dream11 as a Data Scientist
10. Have mentored working professions in field of Data Science and Analytics
11. Have been mentoring GATE aspirants to secure a great rank in limited time
12. Have got around 27.5K followers on Linkedin where I share my insights and guide students and professionals.

Telegram Link for Aditya Jain sir:

https://t.me/AdityaSir_PW

[NAT]

#Q. Maximum Subarray Sum – Kadane's Algorithm

Problem Statement: Given an array arr[], the task is to find the subarray that has the maximum sum and return its sum.

Examples

Example 1:

Input: arr[] = {2, 3, -8, 7, -1, 2, 3}

Output: 11

Explanation: The subarray {7, -1, 2, 3} has the largest sum 11.

Example 2:

Input: arr[] = {-2, -4}

Output: -2

Explanation: The subarray {-2} has the largest sum -2.

Example 3:

Input: arr[] = {5, 4, 1, 7, 8}

Output: 25

Explanation: The subarray {5, 4, 1, 7, 8} has the largest sum 25.

[NAT]

#Q. Find Smallest Missing Positive Number

Problem Statement: Given an unsorted array arr[] containing both positive and negative elements, the task is to find the smallest positive number missing from the array.

Note: You can modify the original array.

Examples

Example 1:

Input: arr[] = {2, -3, 4, 1, 1, 7}

Output: 3

Explanation: The positive numbers in sorted order are {1, 2, 4, 7}. The smallest missing positive number is 3.

Example 2:

Input: arr[] = {5, 3, 2, 5, 1}

Output: 4

Explanation: The positive numbers in sorted order are {1, 2, 3, 5, 5}. The smallest missing positive number is 4.

Example 3:

Input: arr[] = {-8, 0, -1, -4, -3}

Output: 1

Explanation: There are no positive numbers in the array, so the smallest missing positive number is 1.

Appl :-

```
int ans = 1
sort(a)
for( i = 0 ⟶ n)
{
    if( a[i] == ans)
    {
        ans++
    }
    else if (a[i] > ans)
    {
        break;
    }
}
```

Appr2 : Standard : using elements as index.

A[N] → [1 ⟶ N]

no

Smallest pos int

N size

[1 ⟶ N+1]

temp = [0] * (max(arr) + 1)

[1, 5, 2]

[0, 0, 0, 0, 0, 0]

N = 3

N+1

[1, 3, 2]

[1 ⟶ N+1]

temp [N] = [0, 0, 0 ... 0]

for(int elem : a)

{

if (elem >= 1 and elem <= n)

{

temp(elem - 1) = 1

}

}

for (i = 0 ⟶ n)

if (temp(i) == 0)

{

ans = i + 1

break;

}

[NAT]

#Q. Third largest element in an array of distinct elements

Problem Statement: Given an array of n distinct integers, the task is to find the third largest element in the array.

Examples

Example 1:

Input: arr[] = {1, 14, 2, 16, 10, 20}

Output: 14

Explanation:

- Largest element: 20
- Second largest element: 16
- Third largest element: 14

Example 2:

Input: arr[] = {19, -10, 20, 14, 2, 16, 10}

Output: 16

Explanation:

[MCQ]

#Q. Finding sum of digits of a number until sum becomes single digit

Problem Statement: Given an integer n, the task is to repeatedly find the sum of its digits until the result becomes a single-digit number.

Examples

Example 1:

Input: n = 1234

Output: 1

Explanation:

1. Step 1: 1 + 2 + 3 + 4 = 10
2. Step 2: 1 + 0 = 1

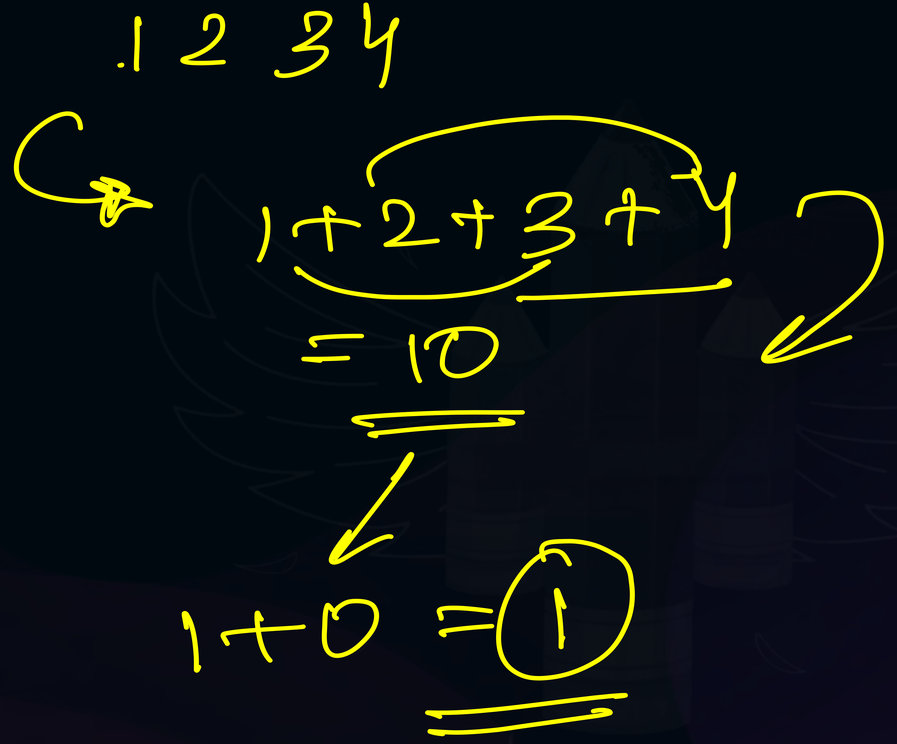
Example 2:

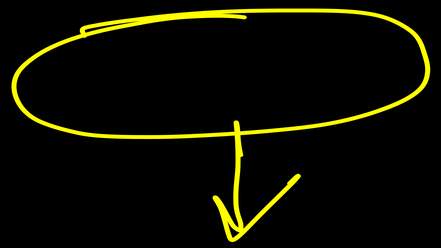
Input: n = 5674

Output: 4

Explanation:

1. Step 1: 5 + 6 + 7 + 4 = 22
2. Step 2: 2 + 2 = 4



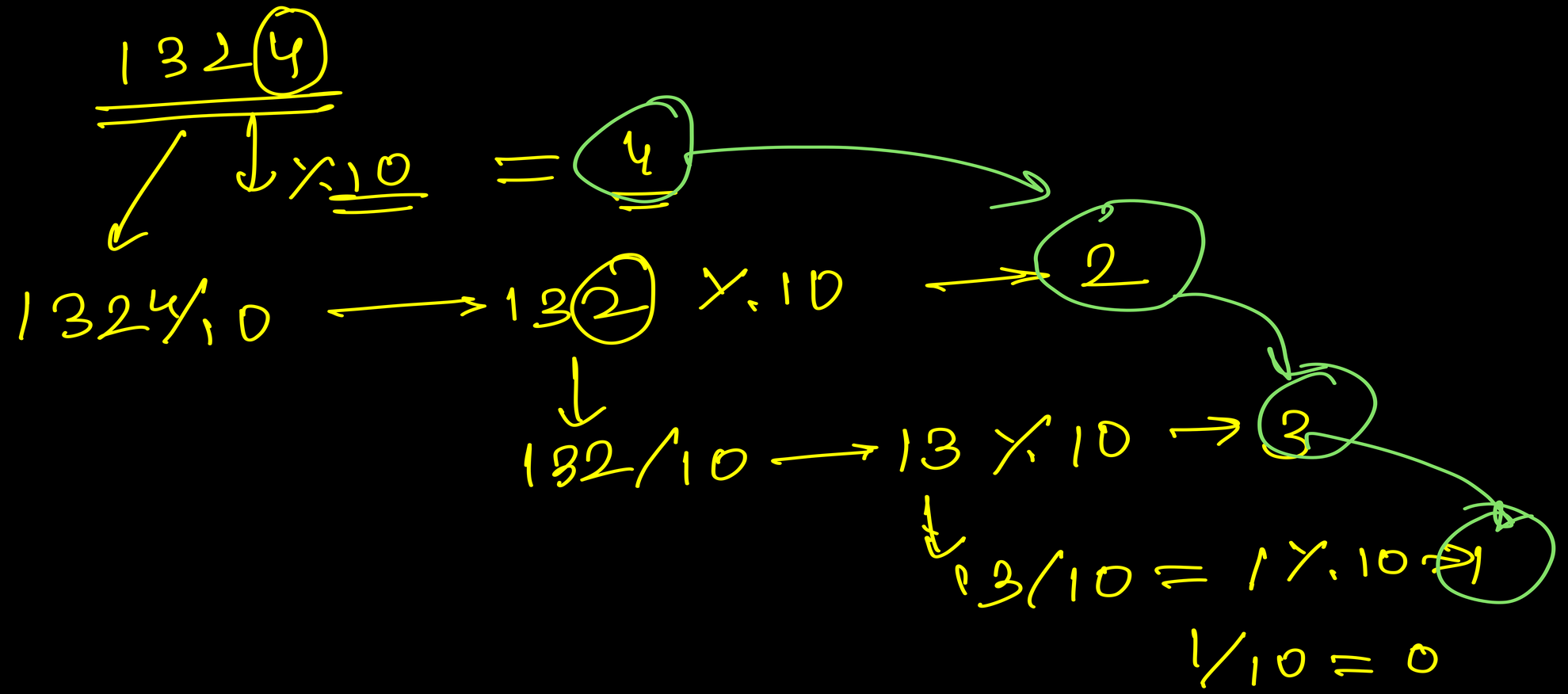


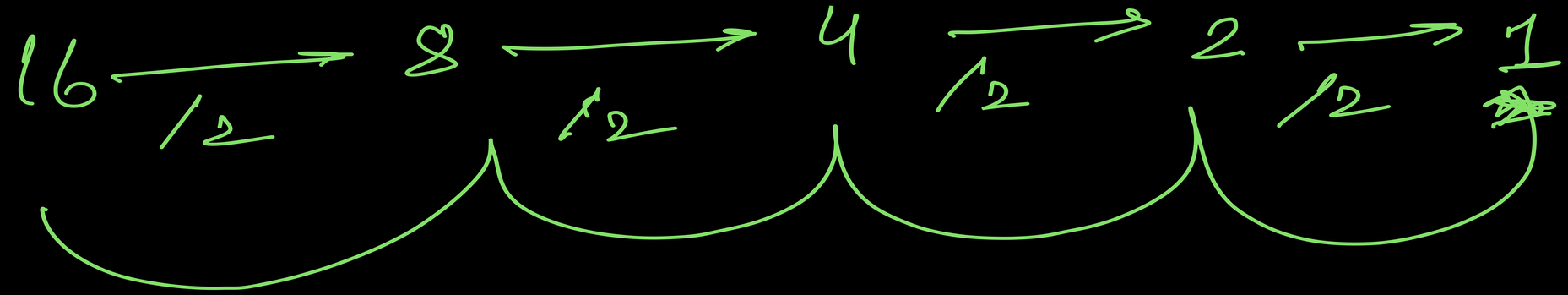
109

$1+0+9 = 10$ ⟶ Stop?

$1+0 = 1$

ans


1324
%10
= 4
1324/10 → 132 %10 → 2
132/10 → 13 %10 → 3
3/10 = 1 %10 → 1
1/10 = 0



$\log_2 n$

10

$\log_{10} n$

[MCQ]

#Q. Sort an array in wave form

Problem Statement: Given an unsorted array of integers, sort the array into a wave array.

An array arr[0..n-1] is in wave form if:

- arr[0] >= arr[1] <= arr[2] >= arr[3] <= arr[4] >= ...

In other words, the elements should alternate between high and low values in a wave-like pattern.

Examples

Example 1:

Input: arr[] = {10, 5, 6, 3, 2, 20, 100, 80}

Output: arr[] = {10, 5, 6, 2, 20, 3, 100, 80}

Explanation:

- The elements follow a wave pattern: large - small - large - small - large - small
- Other valid wave forms could be {20, 10, 6, 2, 100, 3, 80, 5}

Example 2:

Input: arr[] = {20, 10, 8, 6, 4, 2}

Output: arr[] = {20, 8, 10, 4, 6, 2}

Explanation:

- The wave pattern is maintained: large - small - large - small - large - small
- Other valid wave forms could be {10, 20, 6, 8, 2, 4}

[MCQ]

#Q. Longest Common Subsequence (LCS)

Problem Statement: Given two strings, s1 and s2, the task is to find the length of the Longest Common Subsequence. If there is no common subsequence, return 0. A subsequence is a string generated from the original string by deleting 0 or more characters, without changing the relative order of the remaining characters.

For example, sub sequences of “ABC” are “”, “A”, “B”, “C”, “AB”, “AC”, “BC” and “ABC”. In general, a string of length n has 2n sub sequences.

Examples

Example 1:

Input: s1 = "ABC", s2 = "ACD"

Output: 2

Explanation: The longest subsequence common to both strings is "AC", which has a length of 2.

Example 2:

- Input: s1 = "AGGTAB", s2 = "GXTXAYB"
- Output: 4

Explanation: The longest common subsequence is "GTAB", which has a length of 4.

Example 3:

- Input: s1 = "ABC", s2 = "CBA"
- Output: 1

Explanation: The longest common sub sequences of length 1 are "A", "B", and "C". Hence, the result is 1.

[NAT] Imp

#Q. Maximum consecutive one's (or zeros) in a binary array

Problem Statement: Given an array arr[] consisting of only 0's and 1's, the task is to find the maximum count of consecutive 1's or 0's present in the array.

Examples

Example 1:

Input: arr[] = {1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1}

Output: 4

Explanation:

- The longest streak of 1's is 4, from index 8-11.

Example 2:

Input: arr[] = {0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1}

Output: 2

Explanation:

- The longest streak of 0's is 2, from index 0-1.

Example 3:

Input: arr[] = {0, 0, 0, 0}

Output: 4

Explanation:

- The longest streak of 0's is 4, as the entire array consists of 0's.

Ans = 4

int ans = 0 count = 1

for(i = 0 ⟶ n)

{ if (a(i) == a(i-1))

count++;

else

{ ans = max (ans, count)

count = 1;

} }

ans = max (ans, cnt)

Appr2 :- using XOR operator

```
if (a(i) ^ a(i+1) == 0)
{ streak++
}
else {  ans = max(ans, streak)
        streak = 1
     }
```

[NAT]

#Q. Maximum product of a triplet (subsequence of size 3) in array

Problem Statement:Given an integer array, the task is to find the maximum product of any three numbers (triplet) in the array. Examples

Example 1:

Input: arr[] = [10, 3, 5, 6, 20]

Output: 1200

Explanation: The triplet with the maximum product is (10, 6, 20): 10 × 6 × 20 = 1200

Example 2:

Input: arr[] = [-10, -3, -5, -6, -20]

Output: -90

Explanation: The triplet with the maximum product is (-3, -5, -6): (-3) × (-5) × (-6) = -90

Example 3:

Input: arr[] = [1, -4, 3, -6, 7, 0]

Output: 168

max1
max2 } → product
max3

m1 x m2 x m3

Approach :- Brute force :-

All possible triplets

& check product

```
for (i = 0 ⟶ n)
    for (j = i+1 ⟶ n)
        for (k = j+1 ⟶ n)
            maxPro = max (maxPro, a[i] x a[j] x a[k])
```

⟶ $O(n^3)$

maxpro = max1 * max2 * max3

- * - ⟶ +ve
+ * + ⟶ +ve
- * + ⟶ -ve

array ⟶ sort

$A = [-10, -3, -5, -6, -20]$

sort

$[-20, -10, -6, -5, -3]$

$-6 \times -5 \times -3$

$= 30 \times -3 = \underline{\underline{-90}}$

$$A = [1, -4, 3, 7, -6, 0]$$

↓

sort

$$[-6, -4, 0, 1, 3, 7]$$

m3 m2 m1

$-6 \times -4 = 24$

$24 \times 7 = 168$

$1 \times 3 \times 7 = 21$

2

24

7

168

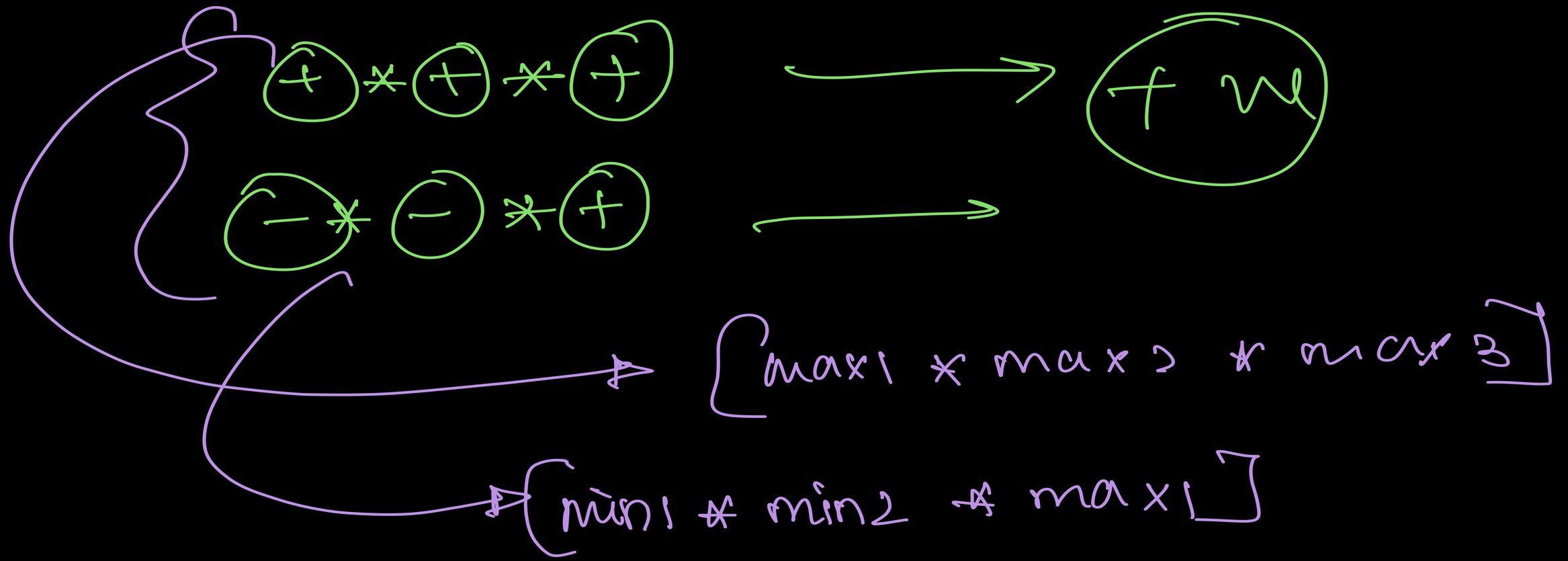
$= 168$

Max 3 elem product

X

Max prod of 3 elements

Sorted : $[a_0\ a_1\ \cdots\ a_{n-1}]$



THANK - YOU