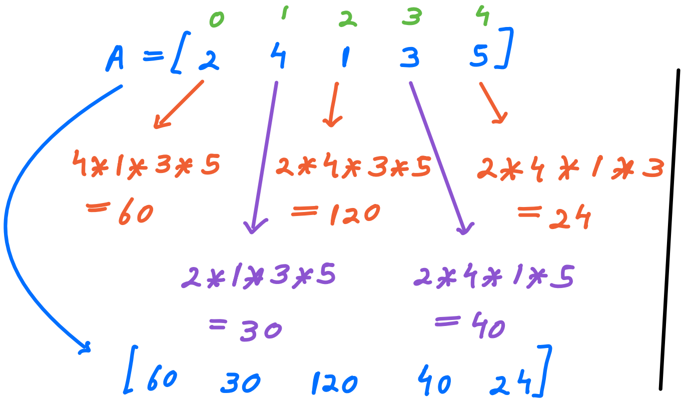
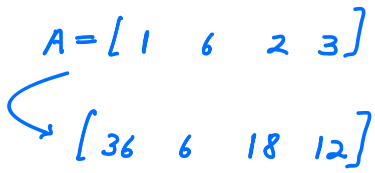
Q → Given an integer array, replace every element A[i] with product of all array elements except itself.





Sol1 →

```
p = 1
for i ⟶ 0 to (N-1)
    p *= A[i]
for i ⟶ 0 to (N-1)
    A[i] = p/A[i] ⟶ If(A[i] == 0) ⟶ error!
```

TC = O(N)
SC = O(1)

No overflow issue.

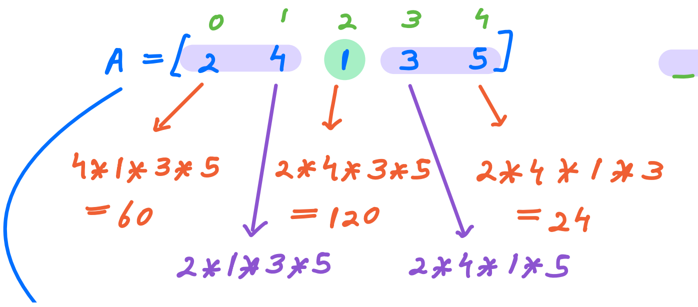
(A[i] == 0) → A[i] → product of all remaining element
All remaining element → 0

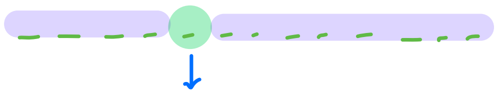
If there are >1 (A[i] == 0) then ∀i A[i] = 0

A = [3 2 0 5]
0 0 30 0

A = [3 0 2 0 5]
0 0 0 0 0

If only '*' is allowed i.e no '/' operator can be used.





A[i] = product of elements from index 0 to (i-1)
*

$$[\overset{}{60} \quad \overset{=30}{30} \quad 120 \quad \overset{=40}{40} \quad 24]$$

product of elements from index (i+1) to (N-1).

prefix product $P[i] = A[i] * P[i-1]$

suffix product $S[i] = A[i] * S[i+1]$ ✓

$(A[i+1] * A[i+2] - - . * A[N-1]) \longrightarrow P[N-1] / P[i]$

$$\begin{array}{lccccc}
 & 0 & 1 & 2 & 3 & 4 \\
A = [& 2 & 4 & 1 & 3 & 5] \\
P = [& 2 & 8 & 8 & 24 & 120] \\
S = [& 120 & 60 & 15 & 15 & 5] \\
[& 60 & 30 & 120 & 40 & 24]
\end{array}$$

```
P[0] = A[0]
for i ⟶ 1 to (N-1)
    P[i] = P[i-1] * A[i]
S[N-1] = A[N-1]
for i ⟶ (N-2) to 0
    S[i] = S[i+1] * A[i]
A[0] = S[1]
A[N-1] = P[N-2]
for i ⟶ 1 to (N-2)
    A[i] = P[i-1] * S[i+1]
```

A = [8]
↓
[1]

$TC = \underline{O(N)}$ $\quad SC = \underline{O(N)}$

```
S[N-1] = A[N-1]
for i ⟶ (N-2) to 0
    S[i] = S[i+1] * A[i]

p = A[0]
A[0] = S[1]
for i ⟶ 1 to (N-2)
    A[i] = p * S[i+1]
    p *= A[i]
A[N-1] = p
```

Q → Given a binary array A[], A[i] → 0, 1

Find max consecutive 1's we can get by replacing atmost one 0 with 1.
{0,1} times

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

4   6 (Ans)   4

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

4 (Ans)   2   3   3

$A = [0\ 0\ 0\ 0\ 0]$ (indices 0 1 2 3 4)

Ans = 1

✓{ $A = [1\ 1\ 1\ 1\ 1]$ (indices 0 1 2 3 4)

Ans = 5

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

4   6 (Ans)   4

(0+1+3)   (3+1+2)   (2+1+1)

```
✓{ if(count of 1's == N)
       return N
ans = 0
for i → 0 to (N-1)
    if(A[i]==0)
        l=0
N ← {   for j → (i-1) to 0
            if(A[j]==0)
                break
            l +=1
        r=0
N ← {   for j → (i+1) to (N-1)
            if(A[j]==0)
                break
            r +=1
        ans = max(ans, l+1+r)
return ans
```

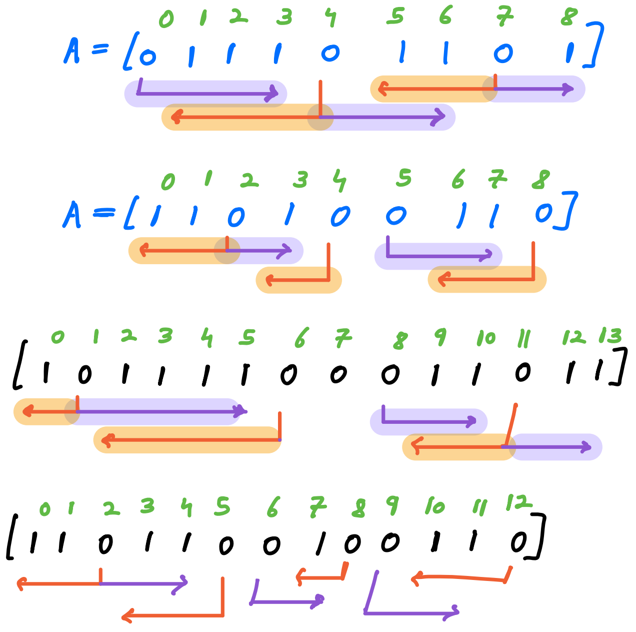
TC = O(N)

SC = O(1)   TC = ~~O(N²)~~ → O(N)

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

for(j = (i-1); j >= 0; j--)

-1



Q → Given a binary array A[] . A[i] → 0

└→ 1

Find max consecutive 1's we can get by ~~replacing~~ → swap atmost one 0 with 1.

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

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

0 1 2 - 1 1 i 1 1 1 ~~1~~ . . . . (N-1)
↓
1 0

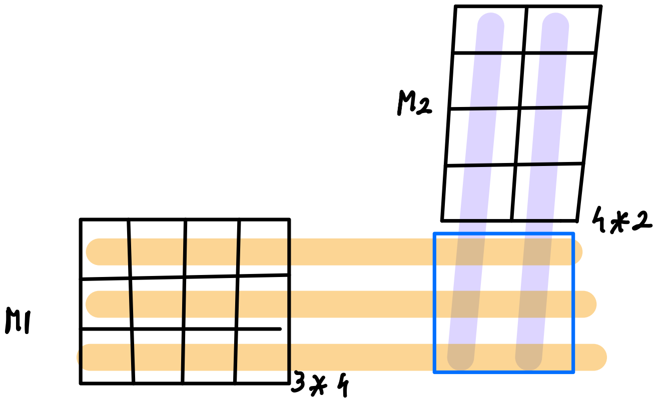
1 0
A = [1 1 ~~0~~ 1 1 ~~1~~ 0]
↓ ↓
6 4

if (ans > count of 1's in A[])
ans = ans-1 // or return count of 1's

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

replace → 6

swap → 5



[1 1 ~~0~~ 1 1]
1

5

[1 1 0 1 ~~1~~]
1 0

4

[1 1 0 1 0 1 1 0 1]
1