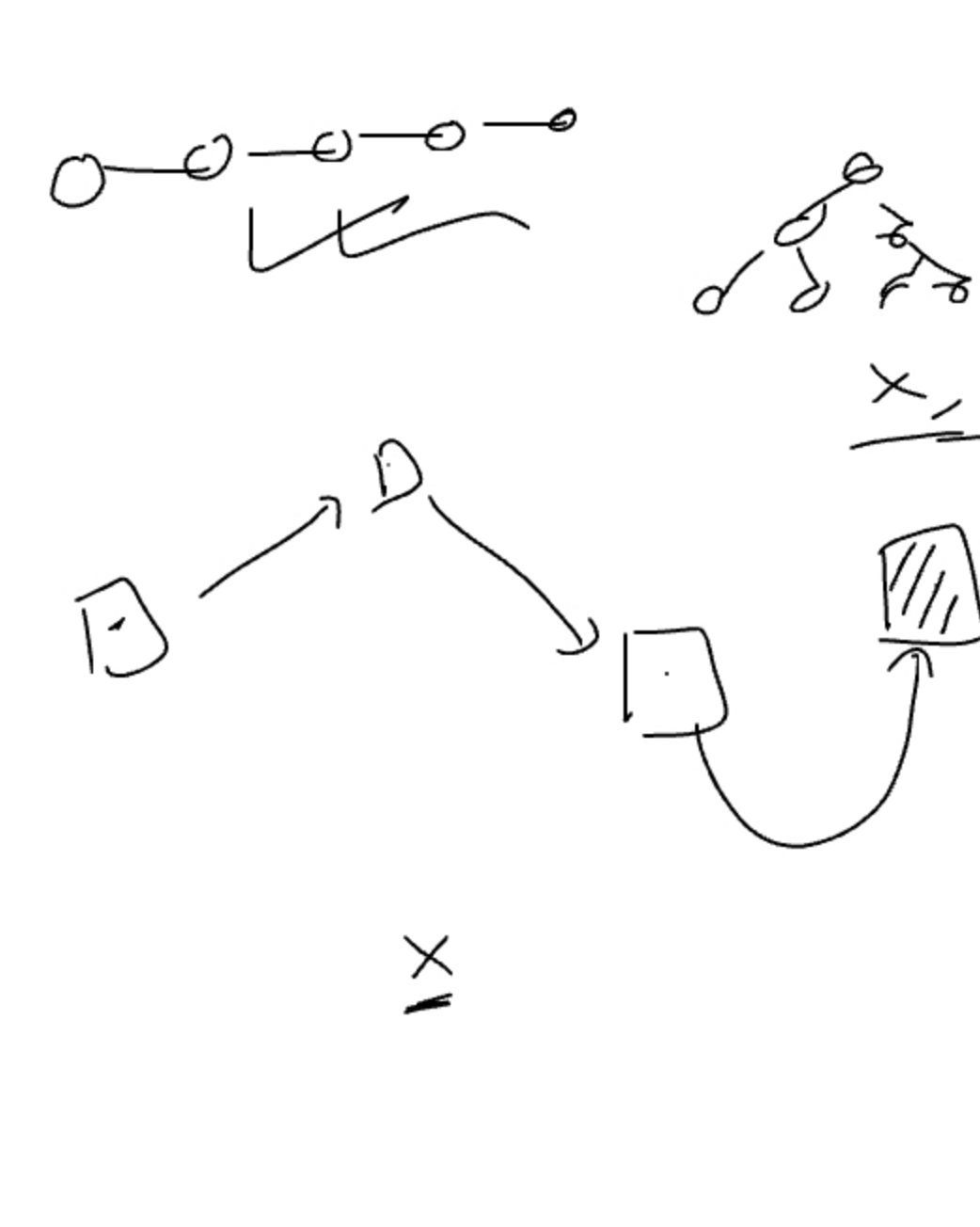
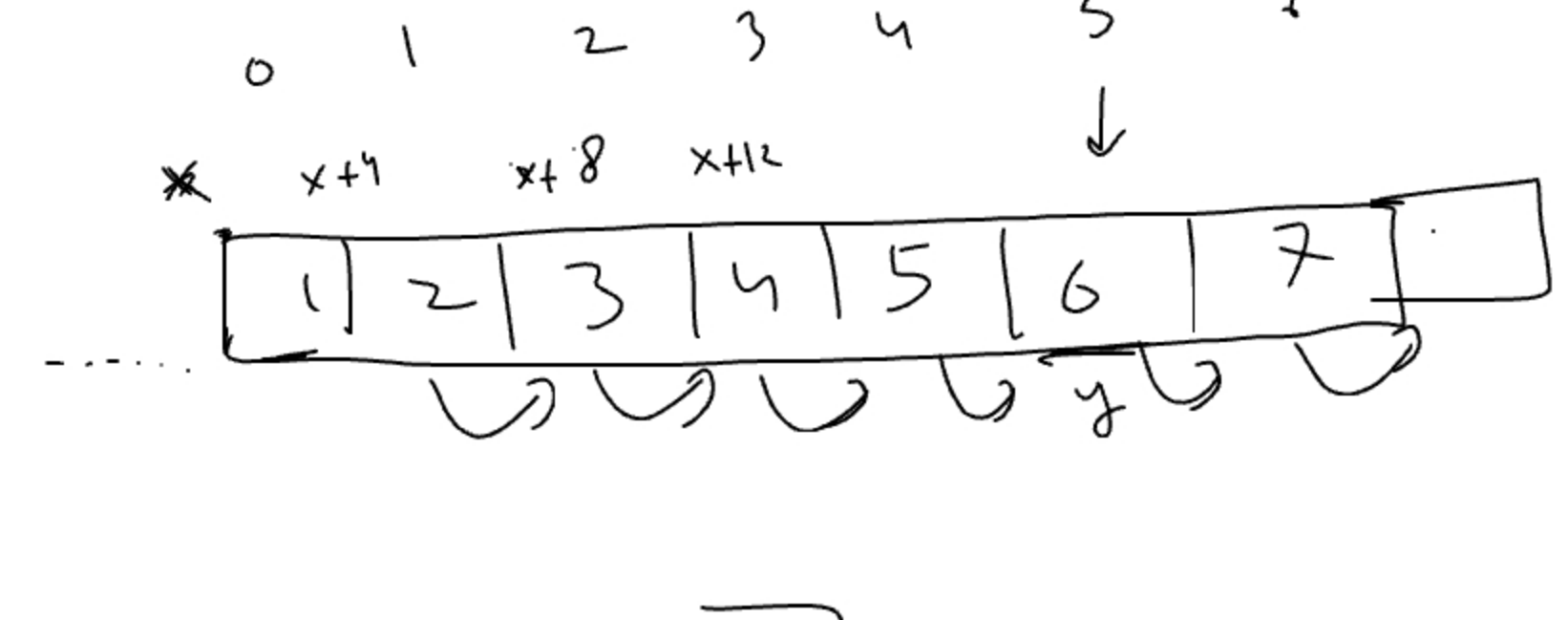


Array - Linked DS.

Insert: $O(n)$.

Contiguous Memory Allocations



- Random Access: $O(1)$
- Insert at end: $O(1)$
- Insert at beginning: $O(n)$
- Delete at end: $O(1)$
- Delete at beginning: $O(n)$

arr: [2, 3, 4]
o/p = 1

arr: [2, 3, 5]

11-9=2.

Simple search: $O(n)$

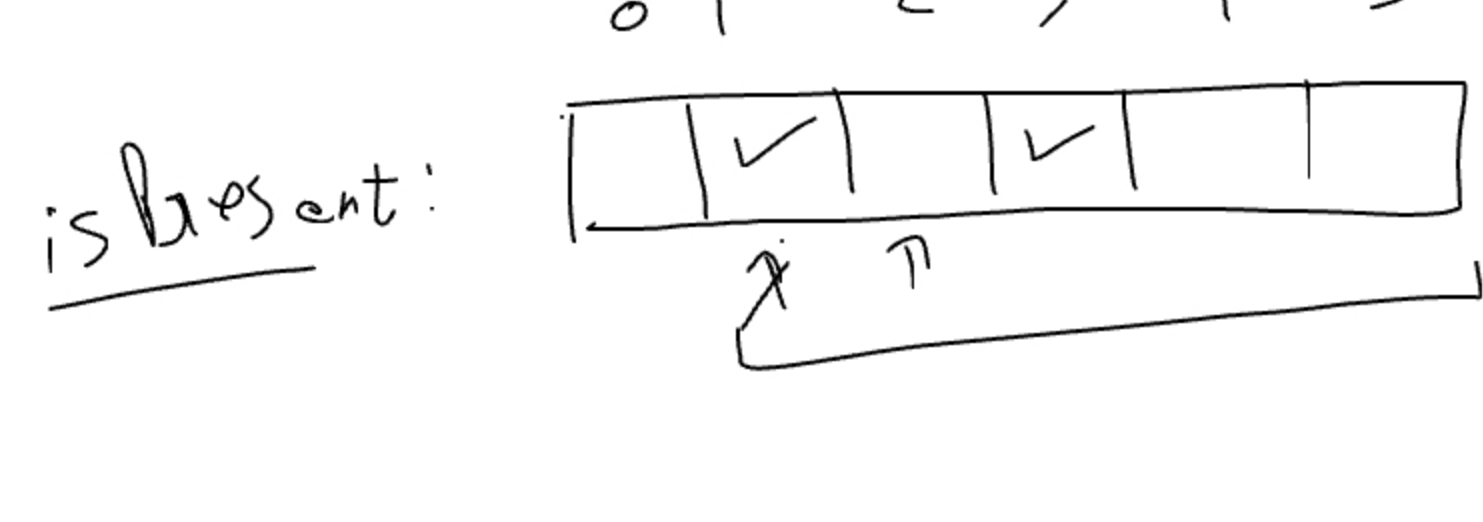
BF: TC: $O(n^2)$

arr: [0, -10, 1, 3, -20] $\xrightarrow{\text{sort}}$ [-20, -10, 0, 1, 3]

TC: $O(n \log n)$

AS: $O(1)$

arr: [0, -10, 1, 3, -20]



TC: $O(n)$

AS: $O(n)$

n \rightarrow res: [1, n+1]

arr: [-2, -5, -3, -2, -4, 1, -7] n=7

Bucket for 1, Bucket for 2, Bucket for 7

TC: $O(n)$

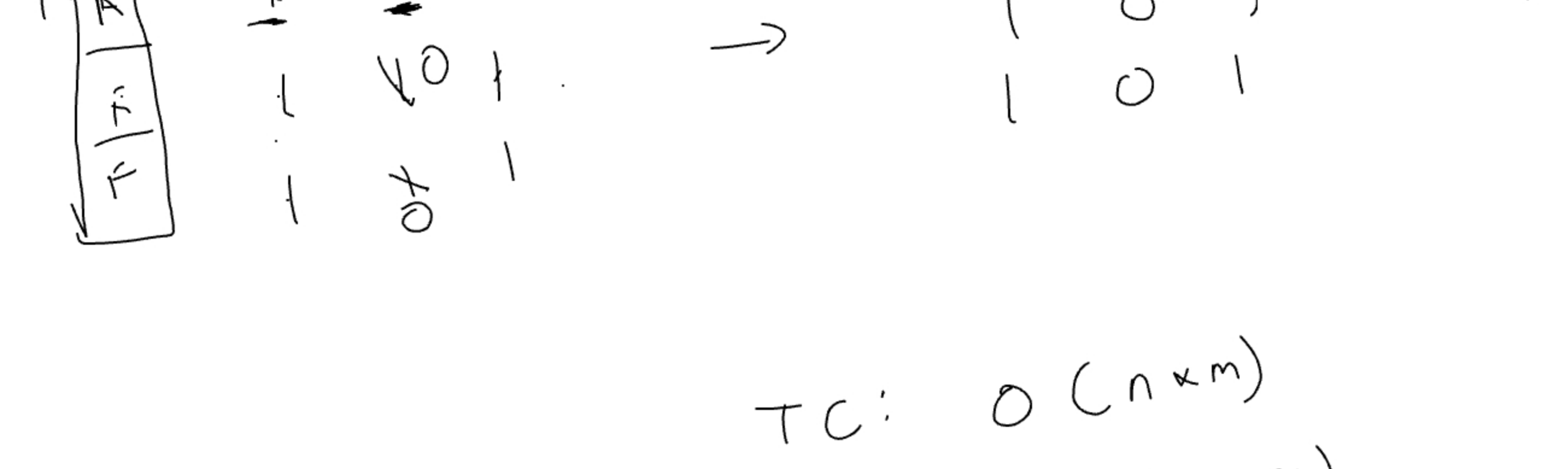
AS: $O(1)$

6 is the bucket for $(i+1)$

arr: [0, -10, 1, 3, -20] n=5

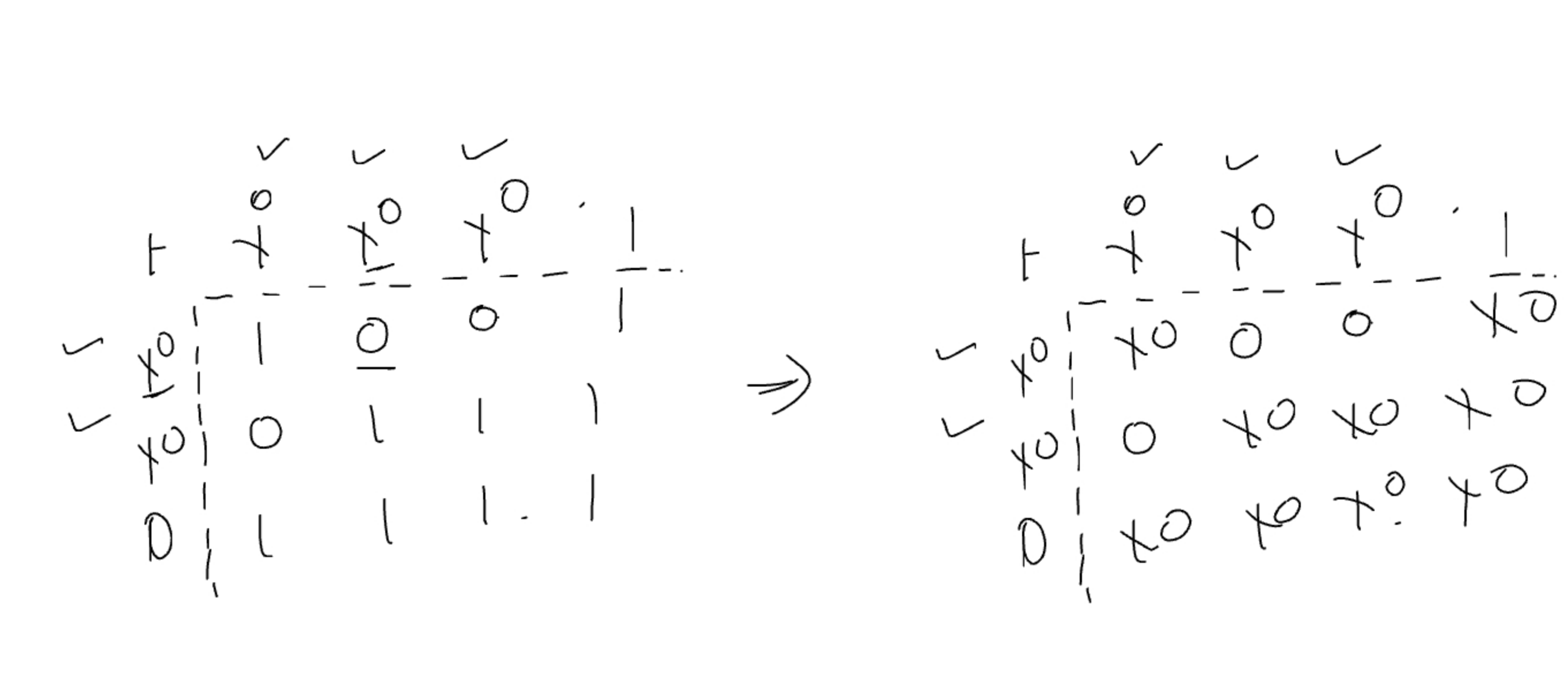
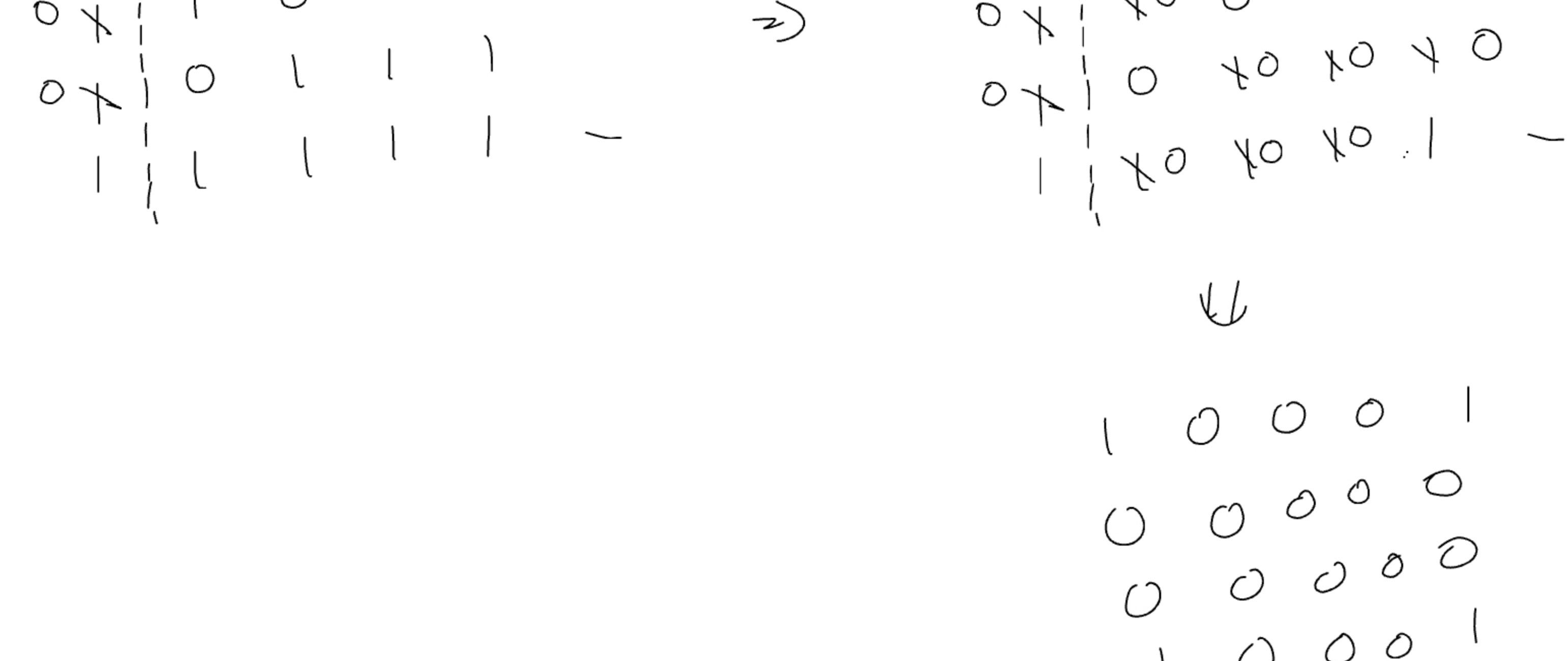
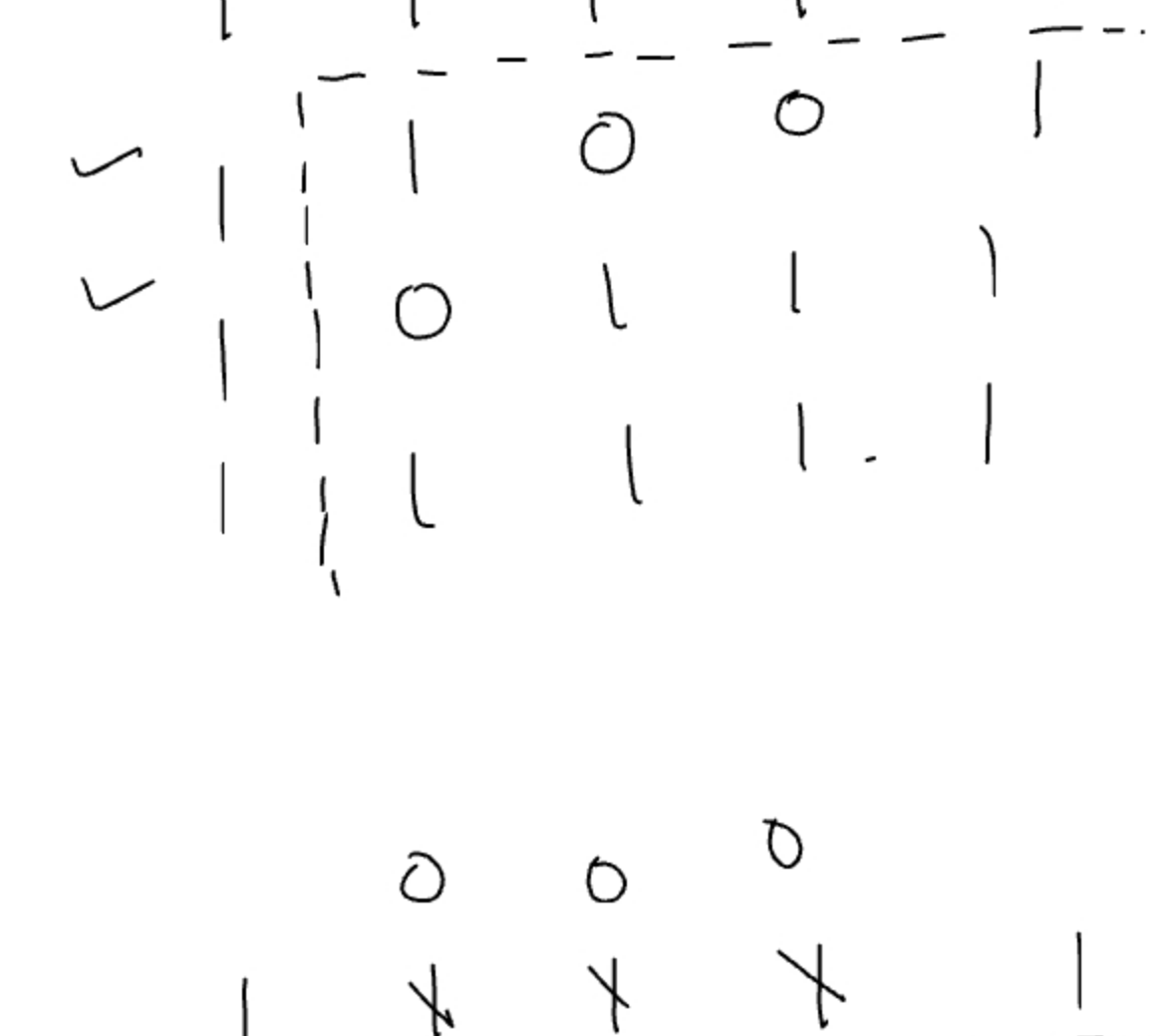
0s & -ves with $n \neq 1$

arr: [-6, 6, -1, 3, 6]



TC: $O(n \times m)$

AS: $O(n+m)$

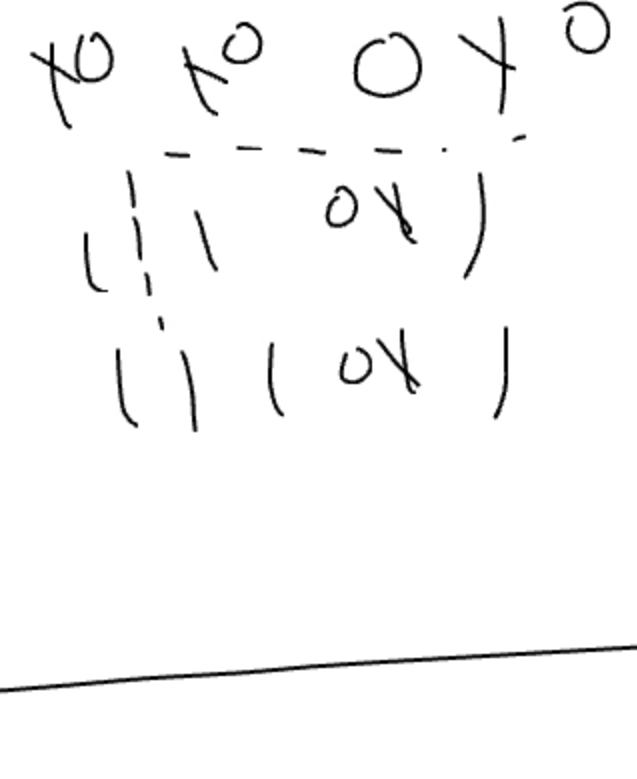


TC: $O(n \times m)$

AS: $O(1)$

frz = true

fcz = false



[2, 3, 5, 1, 7, 8]

Subarray: [2, 3, 5] [5, 1] [7] [1] [5, 8]

[5, 3]

Subsequence: [2, 5] [5, 2] [2, 3, 5], [1], [7]

[1, 2, 3, -2, 5]: [1] [2] [3] [-2] [5]
[1, 2] [2, 3] [3, -2] [-2, 5]

for i in [0, n]
sum = 0
for j in [i, n]
sum += arr[j]
res = max(res, sum)

TC: $O(n^2)$

AS: $O(1)$

maxSum: [1, 2, 3, -2, 5]
1 3 6 4 9

TC: $O(n)$

AS: $O(n)$

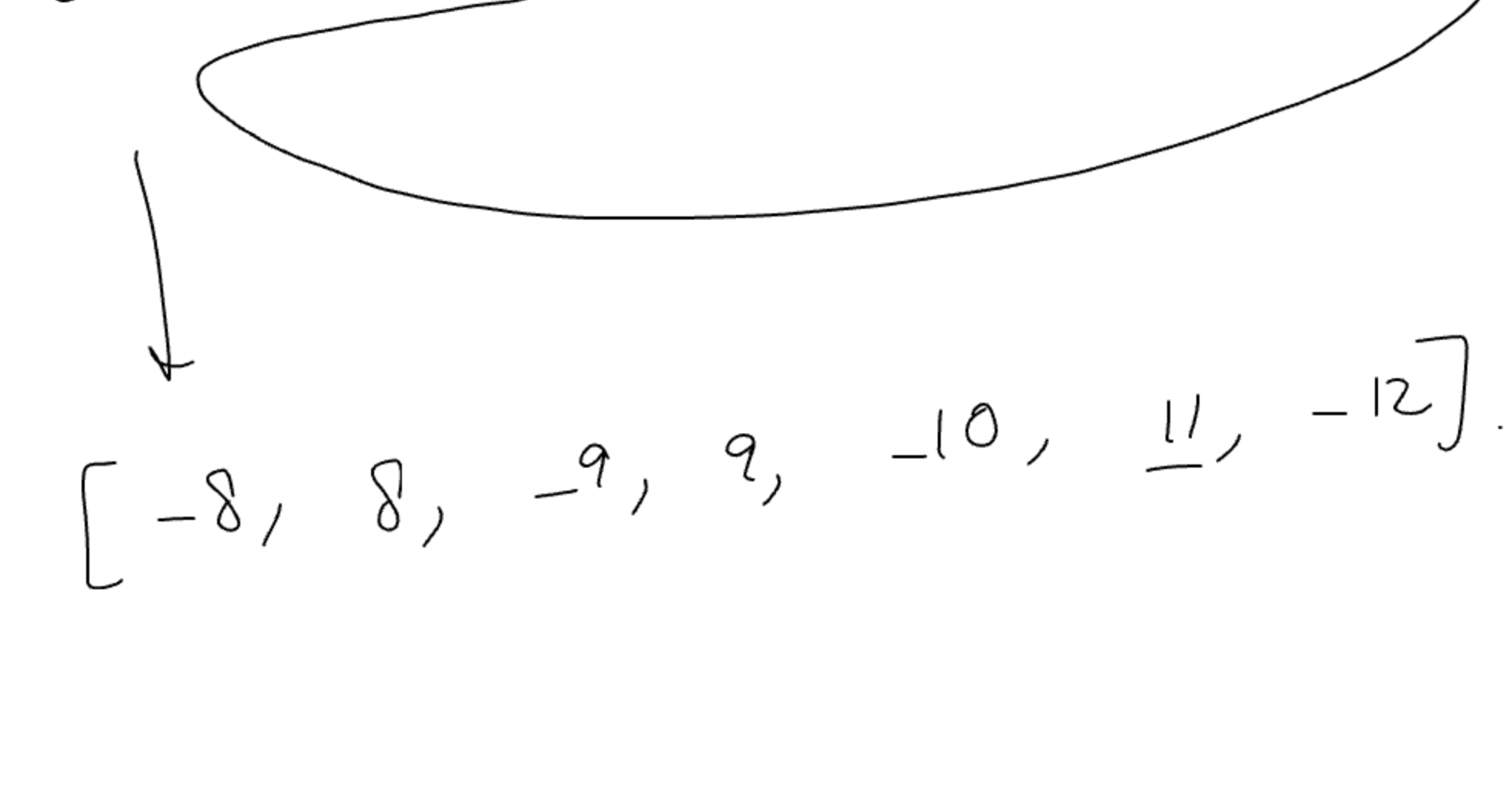
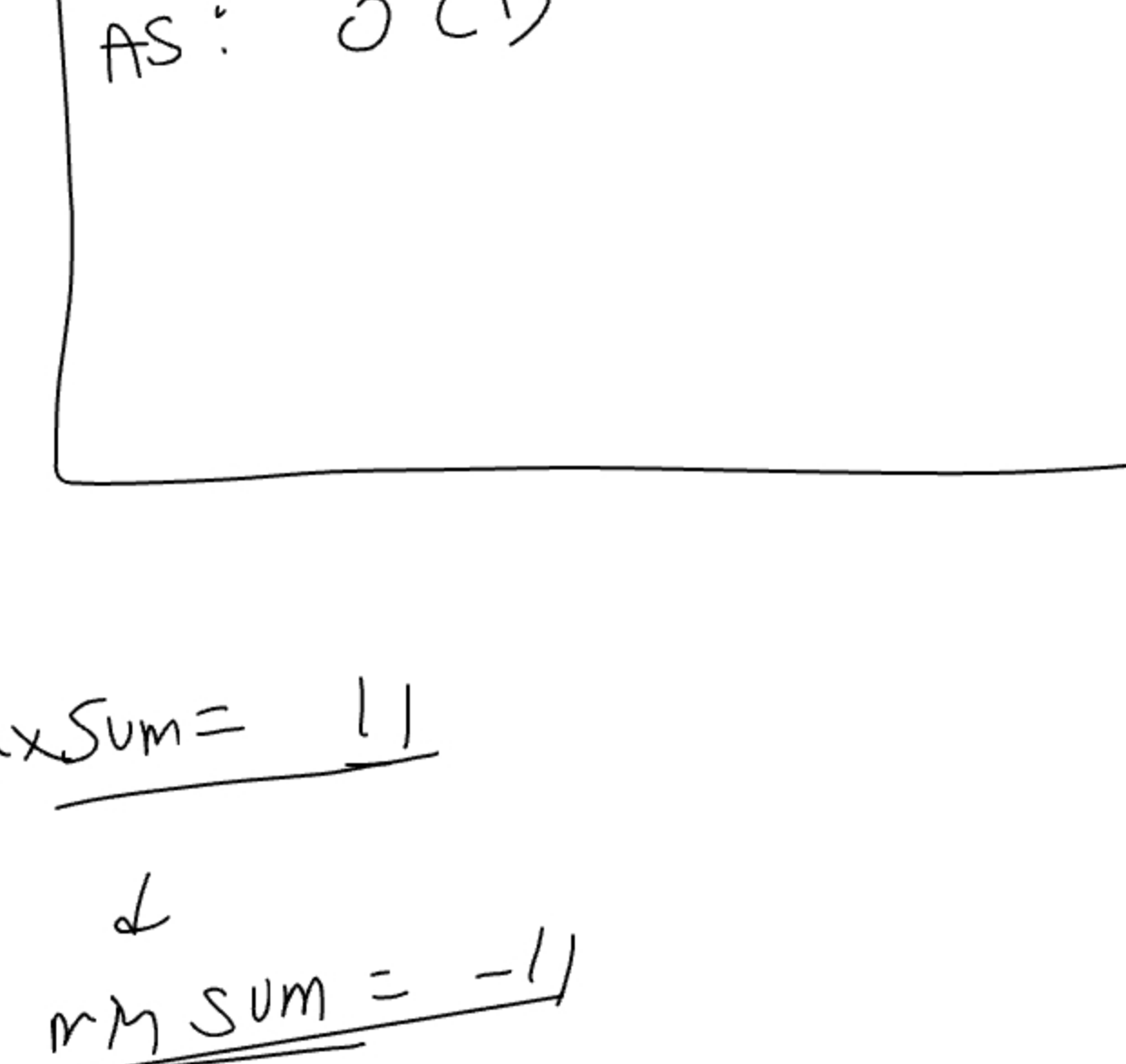
$\text{maxSum}[i] = (\text{arr}[i], \text{maxSum}[i-1] + \text{arr}[i])$

[1, 2, 3] \rightarrow [1, 2, 3, 1, 2, 3]

[-1, 2, 3, -5] \Rightarrow 5

case-1: Circular property not used: Kadane's \rightarrow sum1

case-2: Circular property is used \rightarrow sum2



maxSum = 11

minSum = -11

Overall Sum = 11

sum2 = overallSum - minSum
= 11 - (-11) = 22