Q:- You are given an array of non negative integers. We call any array "special" if for every adjacent pair of elements, the sum is a perfect square.

Return the no. of permutations you can make for the given array that are "special". 2 permutations A1 & A2 differ when there is some index i, such that A1[i] != A2[i]

→ [1, 17, 8] ans → 2
[1, 8, 17] [17, 8, 1]

$1 \le 10$

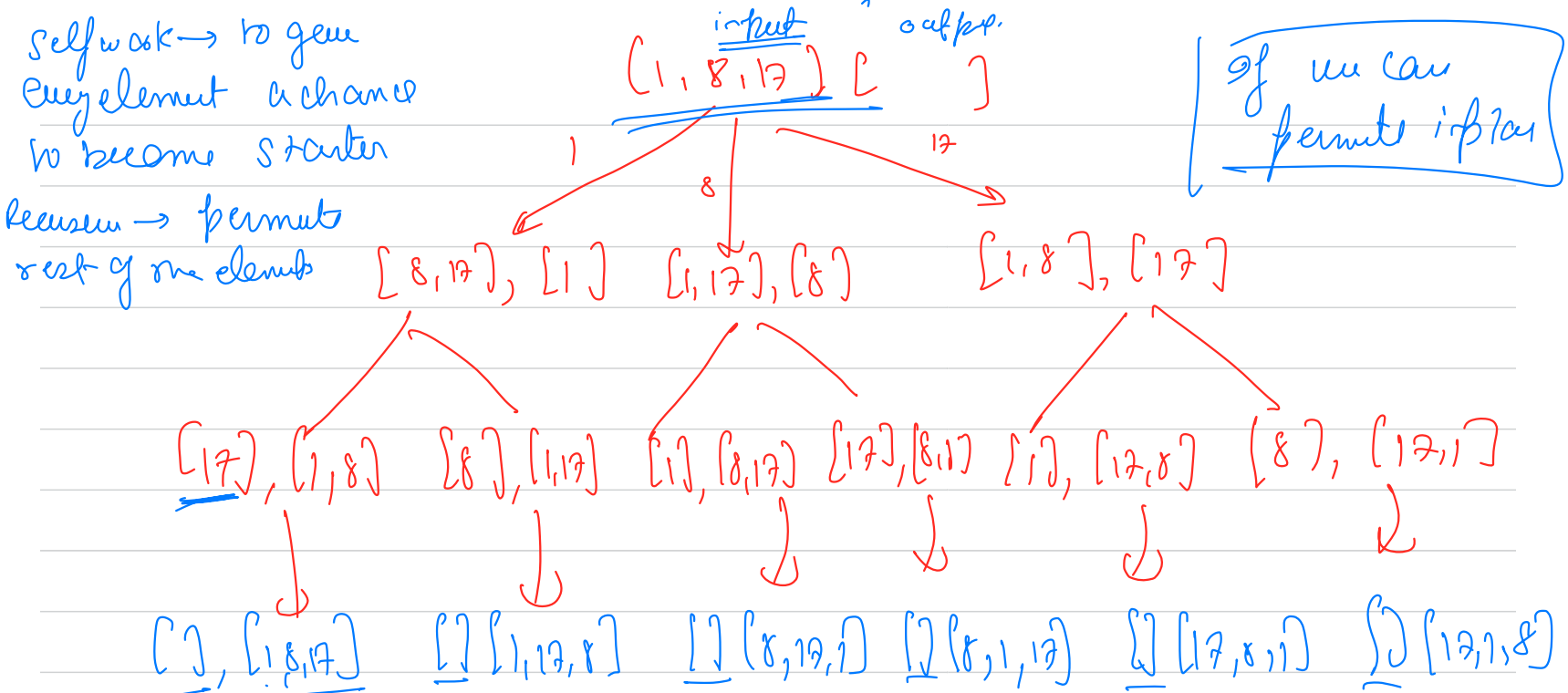
↳ generate all permutations of the given array
↳ to verify if the current permutation is special.

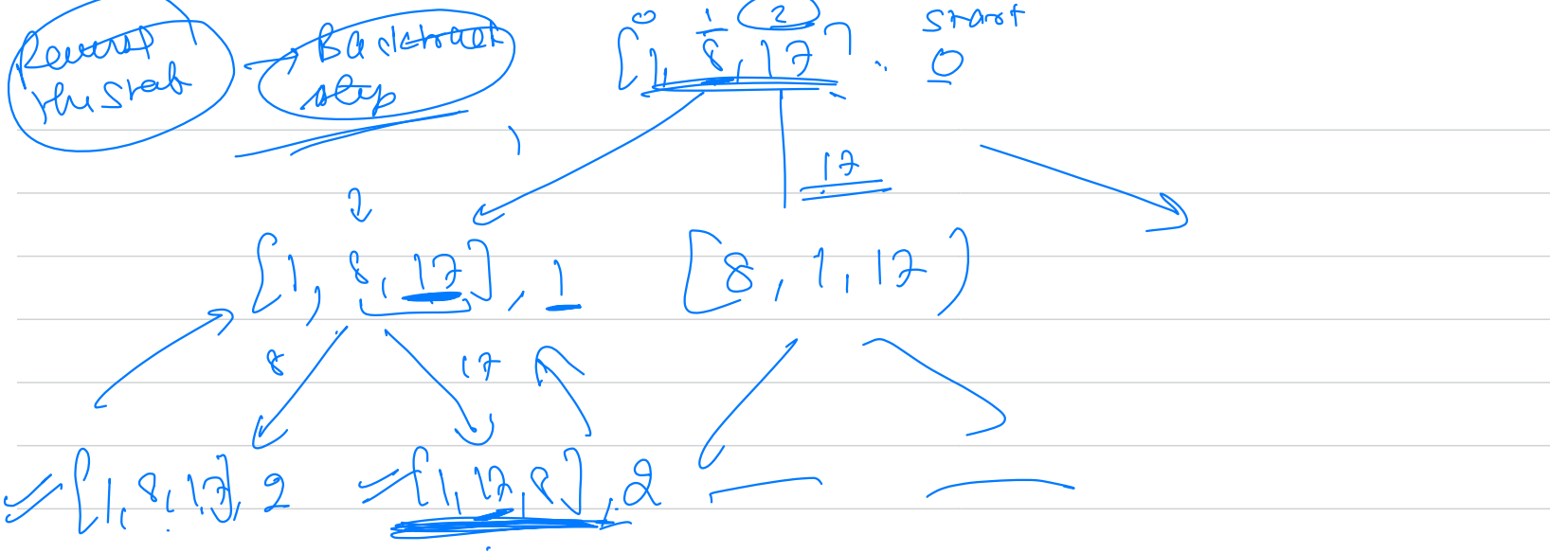
↳ Given an array, calculate all the permutations.

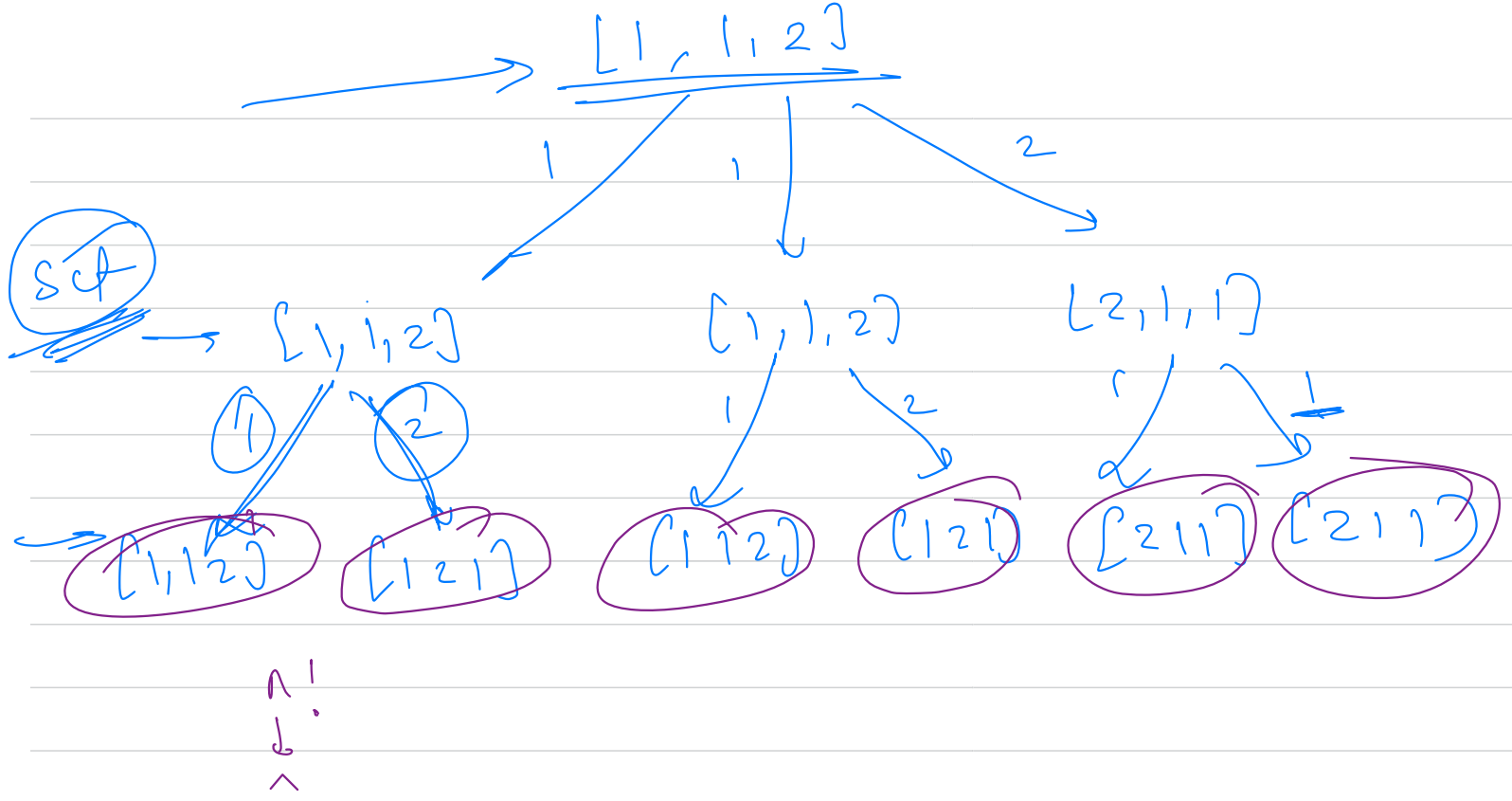
(1, 8, 17) ⟶

1, 8, 17
1, 17, 8
8, 17, 1
8, 1, 17
17, 1, 8
17, 8, 1

everytime, you must fix the first element & let rest of them permute recursively.


selfwork → to give every element a chance to become starter
Recursion → permute rest of the elements
input
output
(1, 8, 17) [ ]
1
8
17
[8,17], [1]
[1,17], [8]
[1,8], [17]
[17], [1,8]
[8], [1,17]
[1], [8,17]
[17], [8,1]
[1], [17,8]
[8], [17,1]
[ ], [1,8,17]
[ ] [1,17,8]
[ ] [8,17,1]
[ ] [8,1,17]
[ ] [17,8,1]
[ ] [17,1,8]
If we can permute it for


Recursive Approach
Backtracking step
[1, 8, 17] : 0
start
2
17
[1, 8, 17], 1
[8, 1, 17)
8
17
[1, 8, 17], 2
[1, 17, 8], 2


[1, 1, 2]
1
1
2
sol
[1, 1, 2]
[1, 1, 2]
[2, 1, 1]
1
2
1
2
1
1
[1, 1, 2]
[1 2 1]
[1 1 2]
[1 2 1]
[2 1 1]
[2 1 1]

$[a_1 | a_2\ a_3 \ldots\ldots a_n]$ ✗ space

(a_1, a_2) ⟶ is not a perfect square

Q> You have an array of integers. We need to sort (sorted) & store in a new array the array elements based on the the square value

[-6, -3, -1, 2, 4, 5]

↳ [1, 4, 9, 16, 25, 36]

[1, 4, 9, 16, 25, 36]

O(n)

$n \leq 10^2$

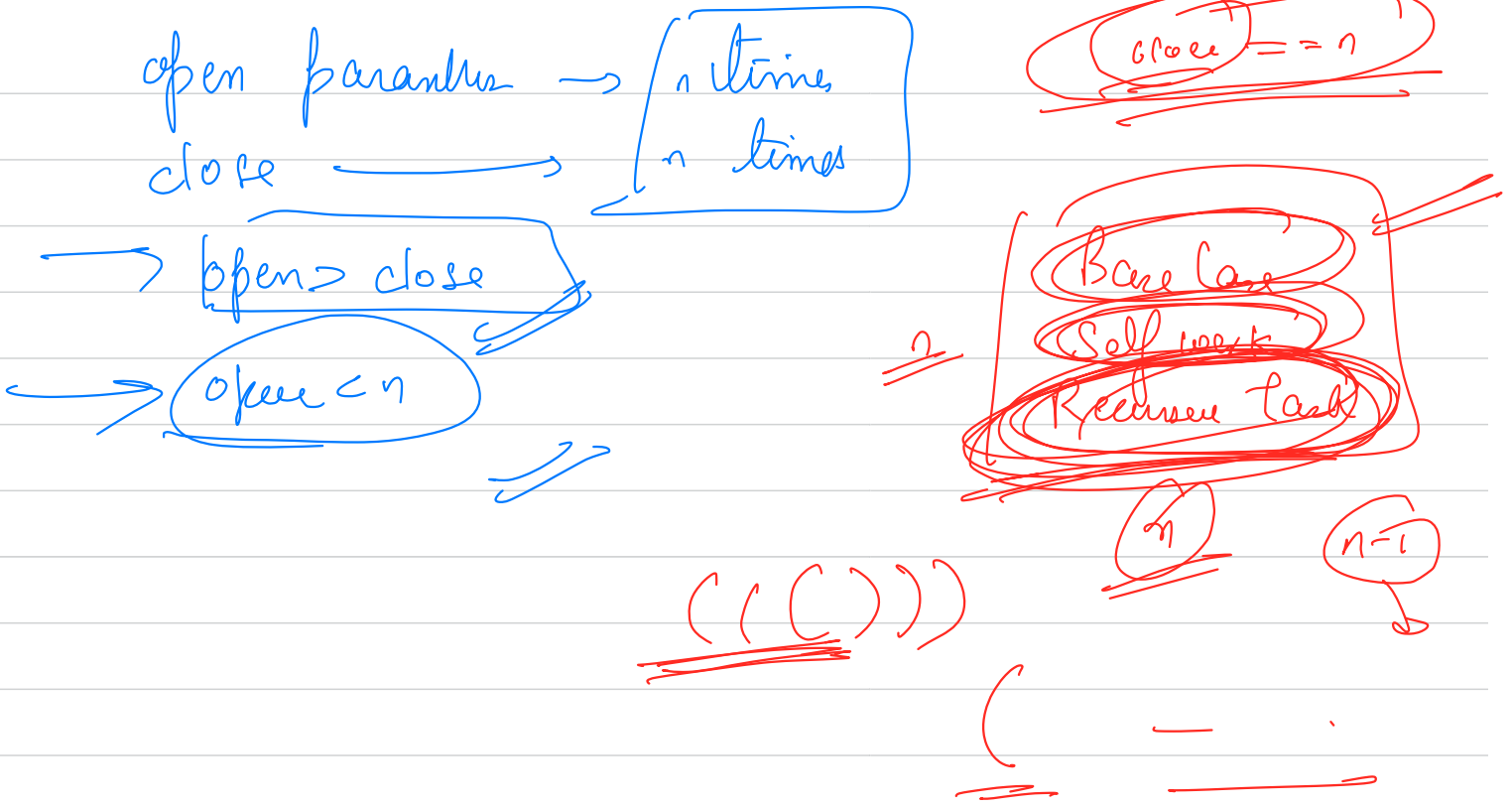
$-10^9 \leq a[i] \leq 10^9$

Q: Given a value of positive integer n, print all combinations of balanced parenthesis of size 2n

n = 1 ⟶ ()

n = 3 ⟶ ((())), ()()(), (())(), ()(()), (()())

$n \leq 20$


open parantes → n times
close → n times
open > close
open < n
close == n
Base Case
Self work
Recursive call
n
n-1
((()))

Q) You are given an array of n, positive elements.

You will get q queries, where in each query you will have 2 number l_i, r_i. You need to find the sum of elements in the range $[l_i, r_i]$.

You need to reorder the original array such that the sum of queries given is maximum possible.

[5, 3, 2] (1 2 3) → 2, 5, 3

q = 3
1 2
2 3
1 3
ans → 25

$1 \le l_i, r_i \le n$

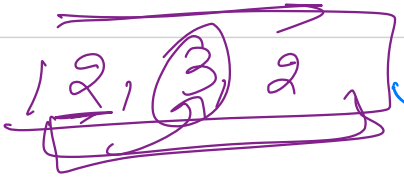
$n \le 10^6$
$q \le 10^6$
$1 \le a_i \le 10^5$

⇒ the index which is queried the most should be mapped to the biggest element for making the sum max

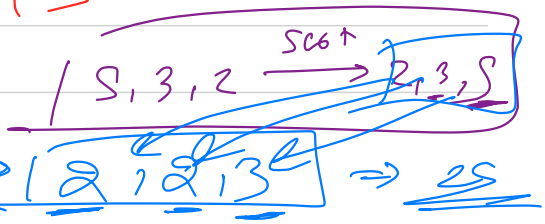
↳ to calculate which index is queried the most →

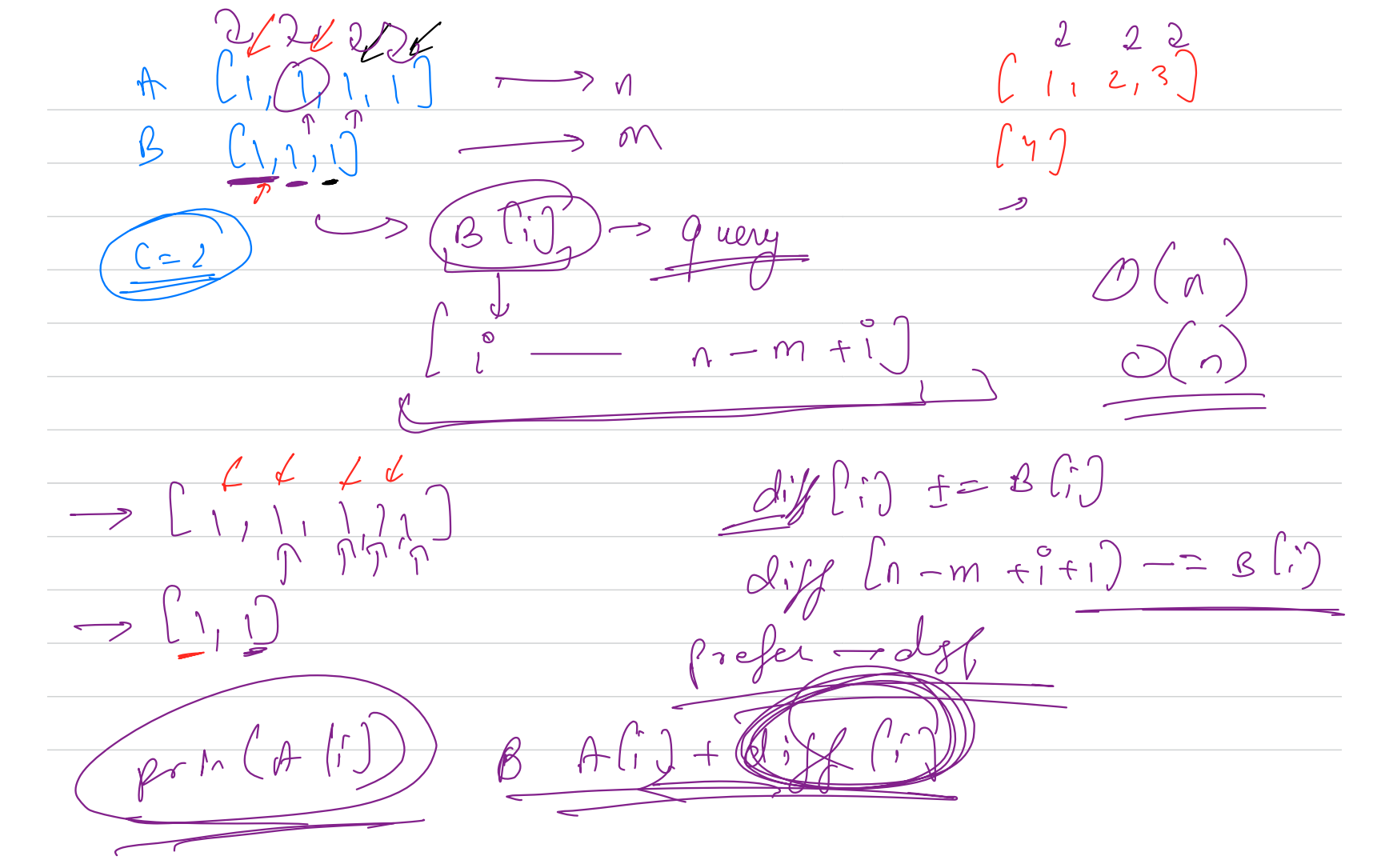
difference array

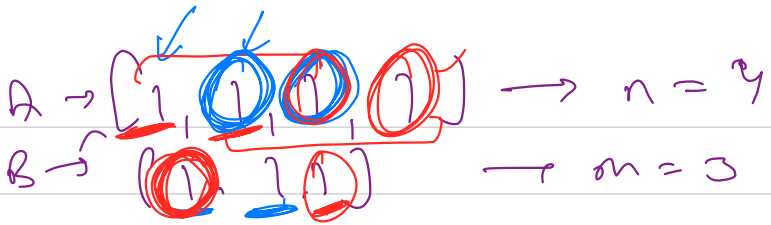
index → [2, 1, -1]
1 2 3



1 2
2 3
1 3




A [1, 1, 1, 1] → n
B [1, 1, 1] → m
C = 2
B[i] → query
[i —— n − m + i]
O(n)
O(n)
[1, 1, 1, 1]
[1, 1]
diff[i] += B[i]
diff[n − m + i + 1] −= B[i]
prefer → dsf
print(A[i])
B A[i] + diff[i]



t = B[i]
— = B[i-1]

prefix → [0, 0, 0, 0]

[1, 0, -1, 0]

[1, 1, -1, -1]

[1, 1, 0, -1]

→ [1, 2, 2, 1]

→ 2 3 3 2

(0 1 1 0)