Data Structures.

↓

Basic operation (DSA)

↓

→ Traversing

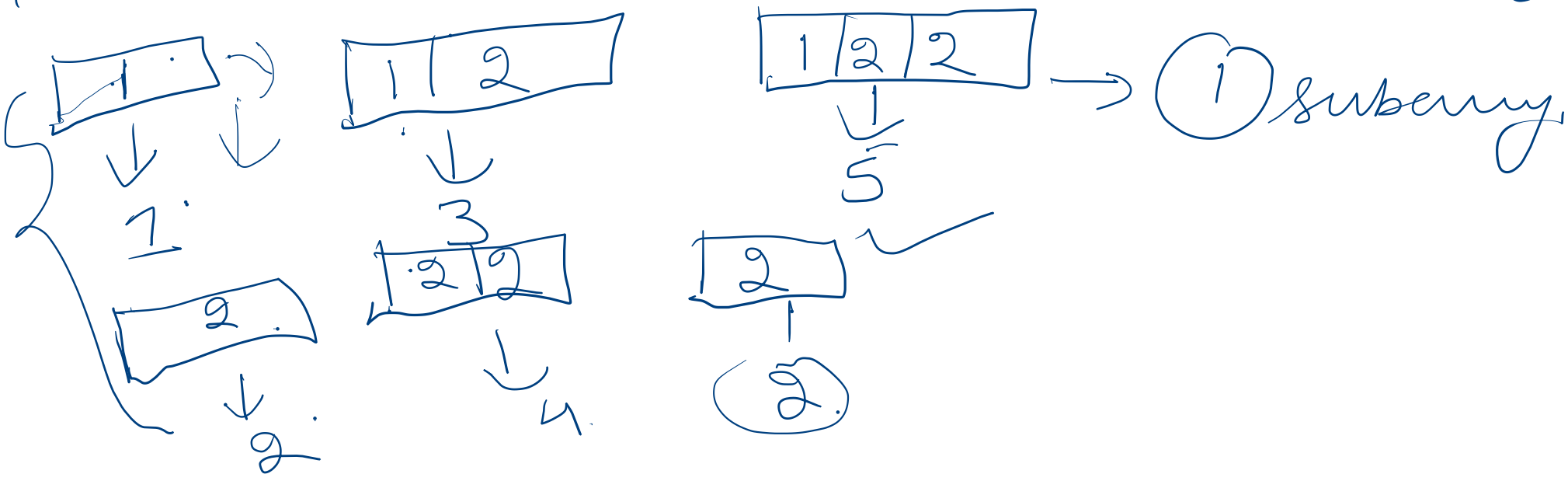
→ Insertion

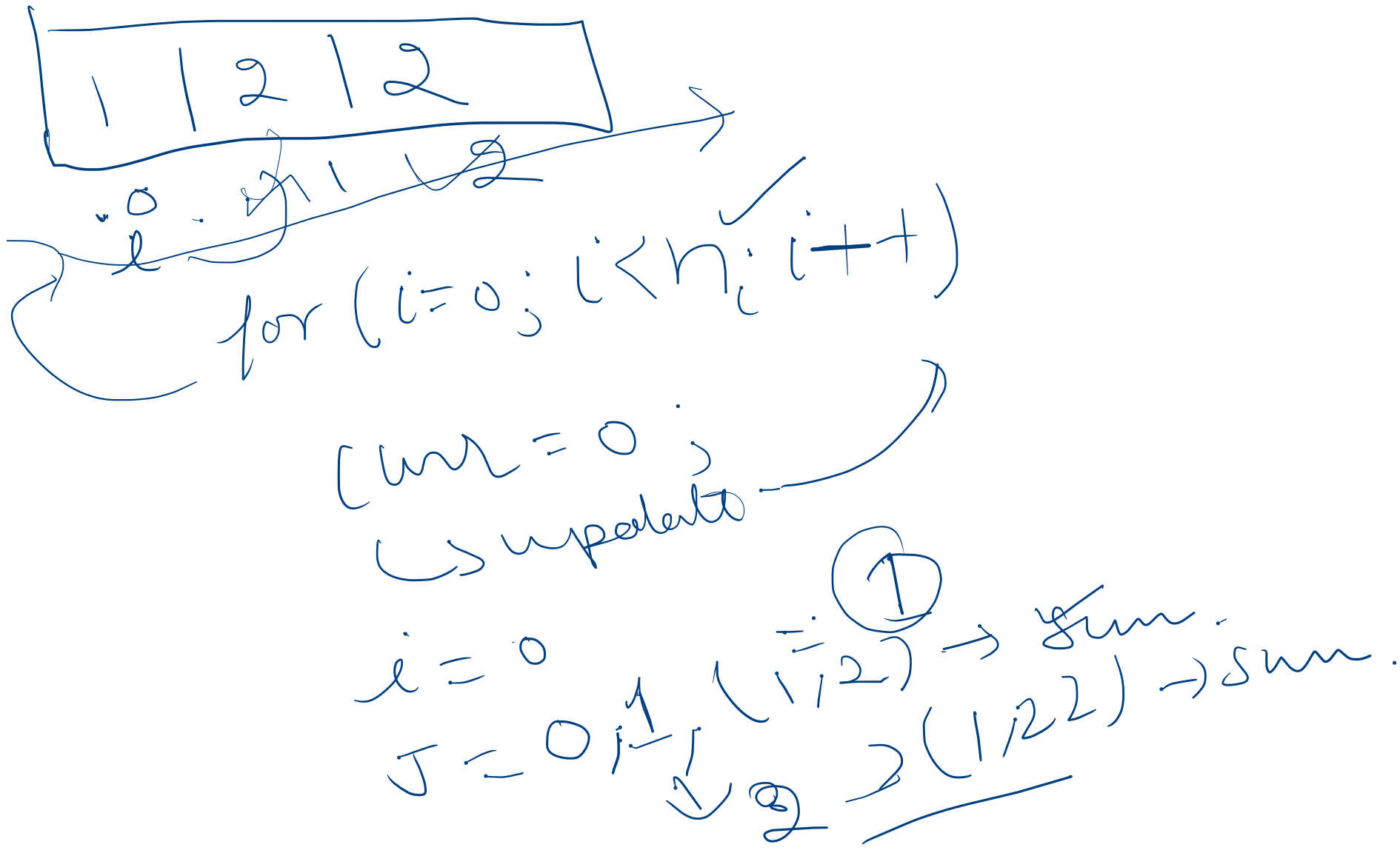
→ Deletion

→ merging

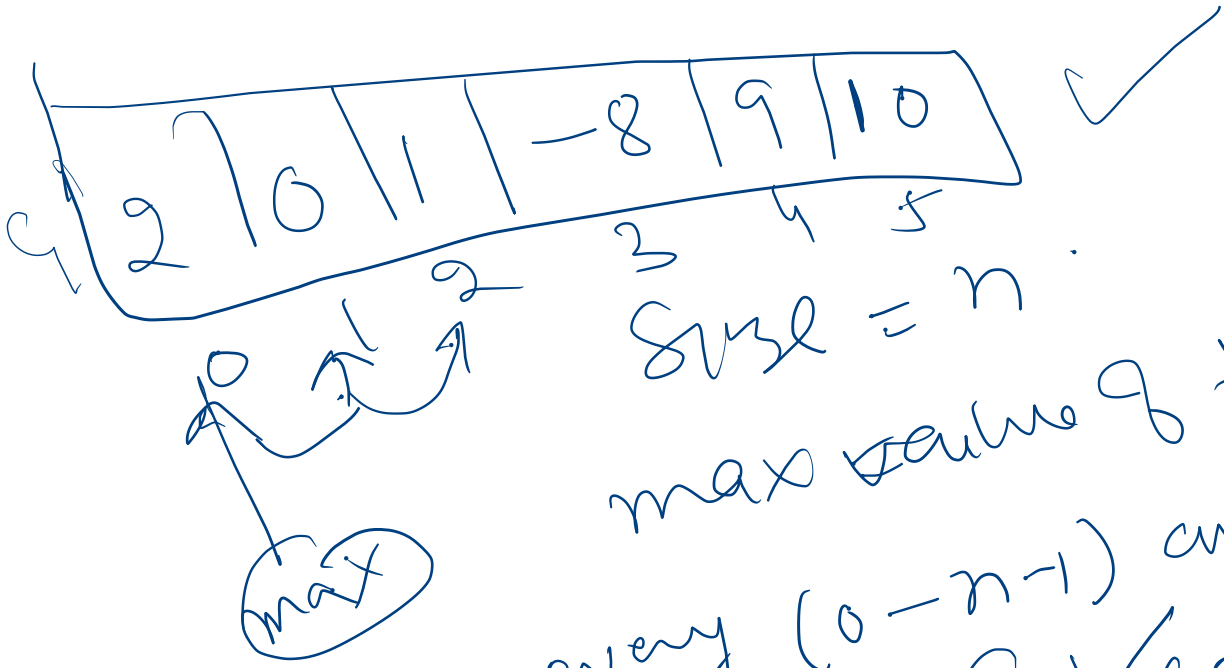
Array Challenge.

- Find out sum of each subarray of the given array.

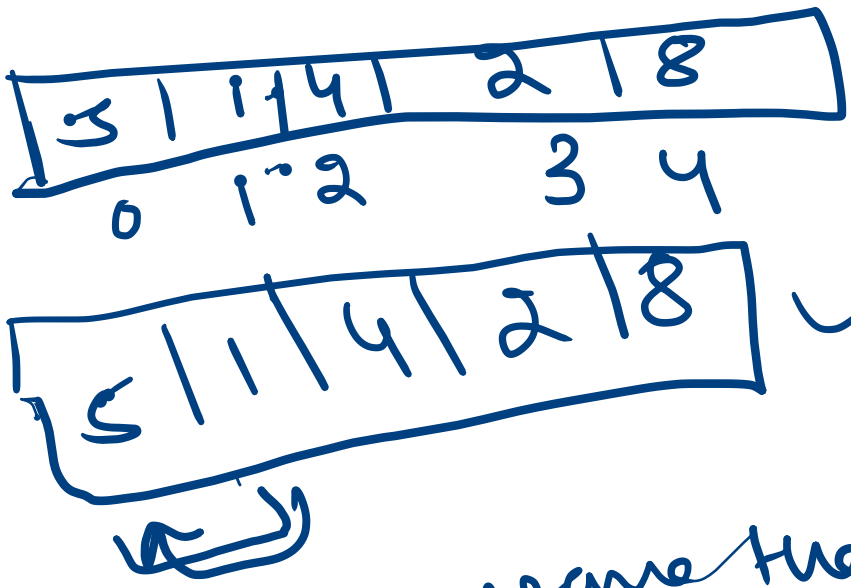


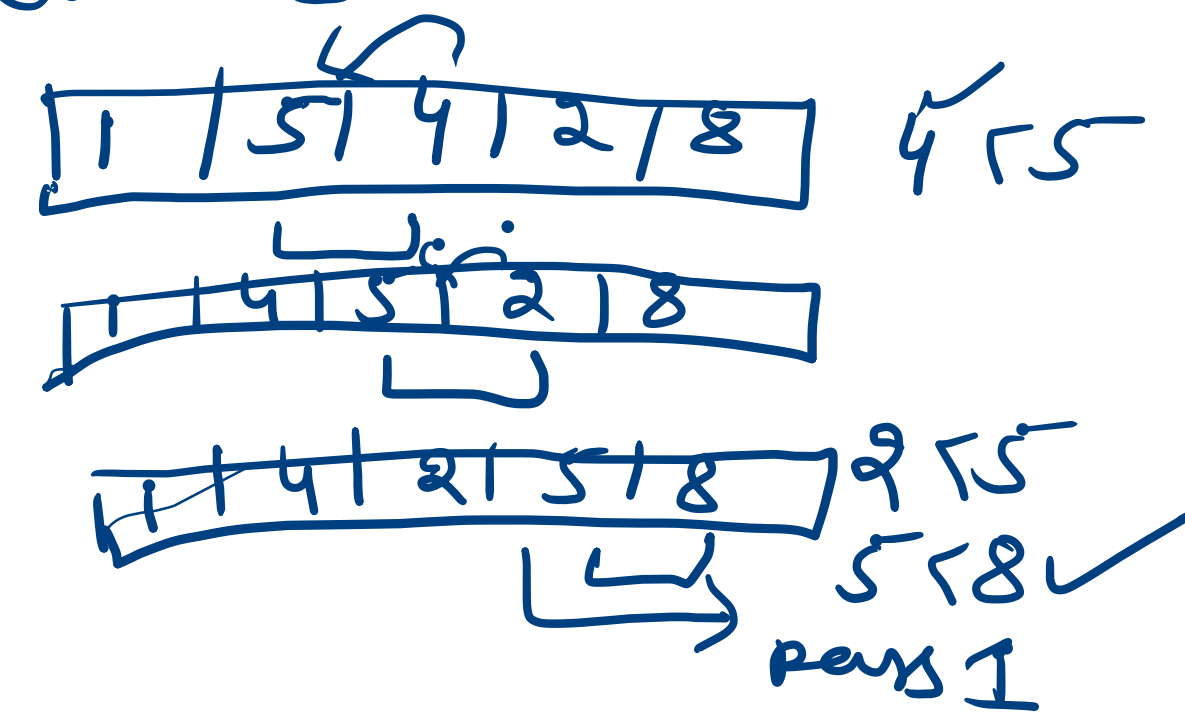

1 1 2 1 2
0 1 2
for (i=0; i<n; i++)
{ sum = 0 ;
i = 0
j = 0 1 2
(1,2) → sum.
(12) → sum.

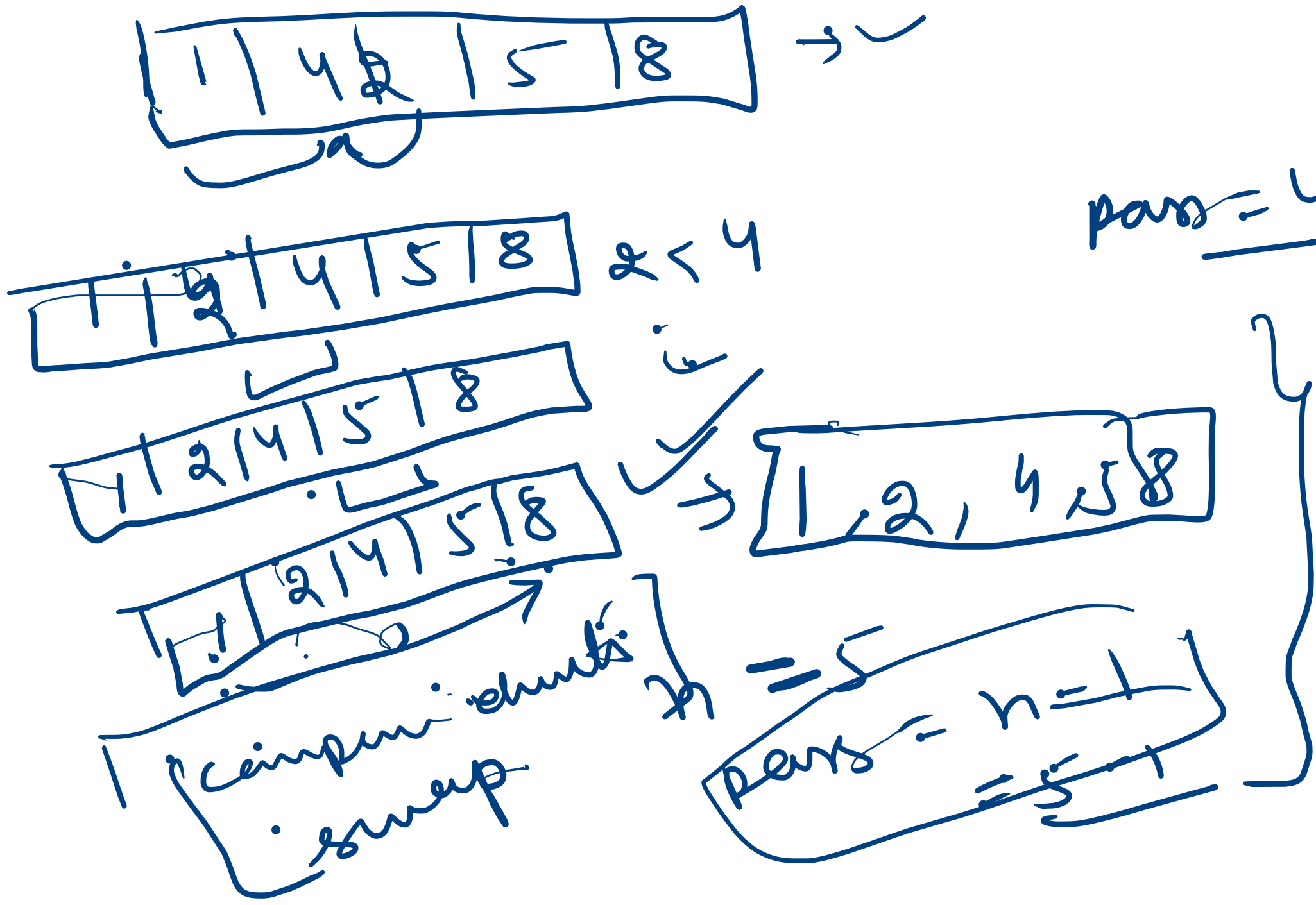
```
int n;

cin >> n;
for (i=0; i<n; i++)
    cin >> arr[i];

for (i=0; i<n; i++)
    for (j=i+1; j<n; j++)
        if (arr[i] > arr[j])
            cout << arr[i];
```

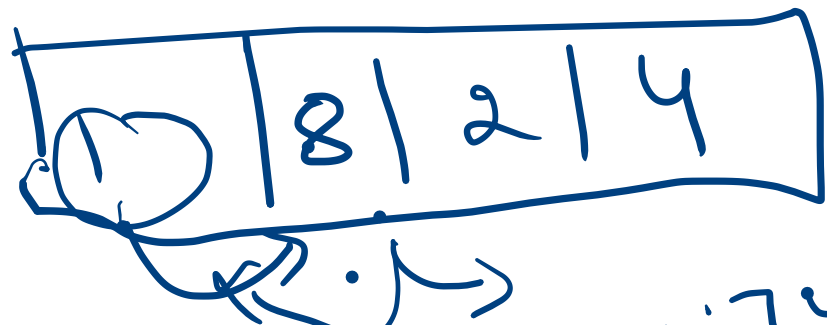

2 10 11 -8 9 10
0 1 2 3 4 5
max
size = n
max value of the array
for every (0-n-1) output should be max
9 (9 > 2)
mx
10 > 9
mx
final output
10
(max = 99
for (i=0 ; i<n ; i++)
mx = max(max(a[i]
max → point
a[1], a[2] cmt(mx)

Bubble sort


5 1 4 2 8
0 1 2 3 4
5 1 4 2 8
Swap
→ Compare the elements.
→ swap
5 > 1 / 1 < 5

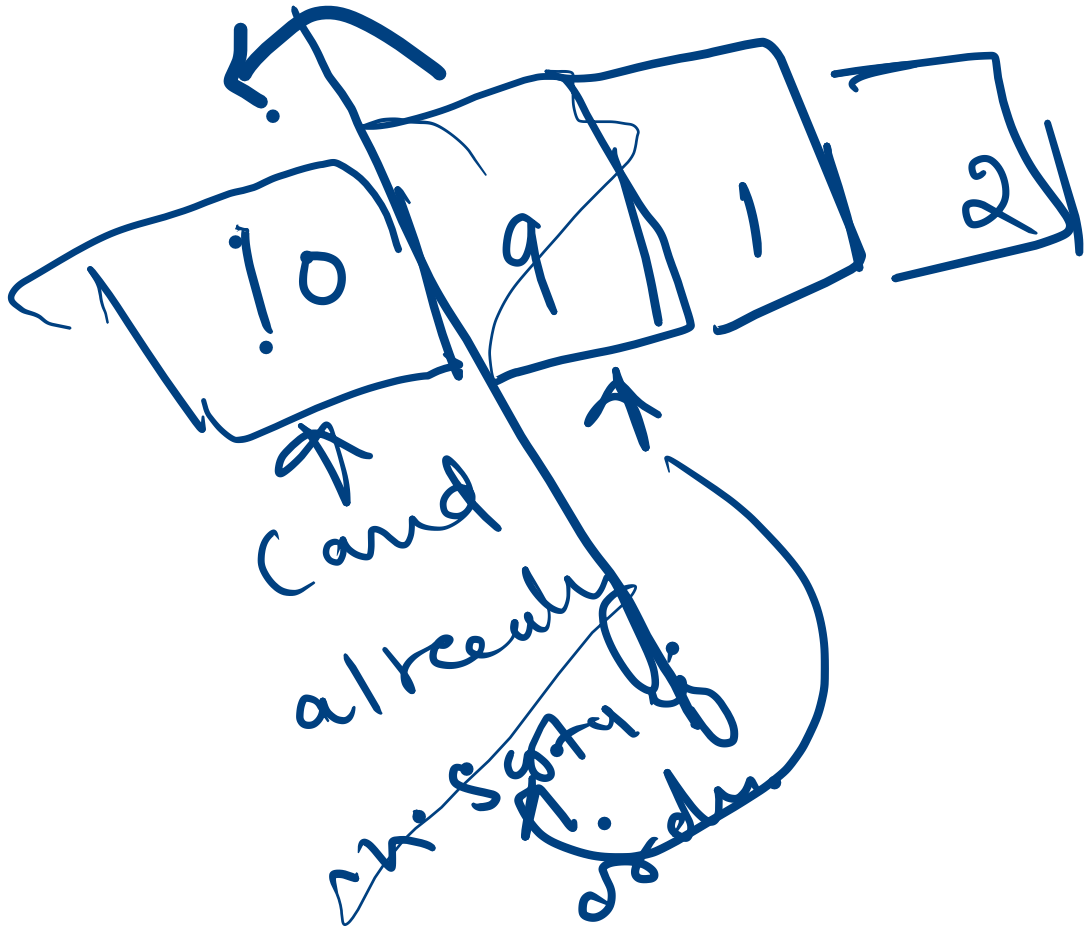

1 5 4 2 8
4 < 5
1 4 5 2 8
1 4 2 5 8
2 < 5
5 < 8
pass 1

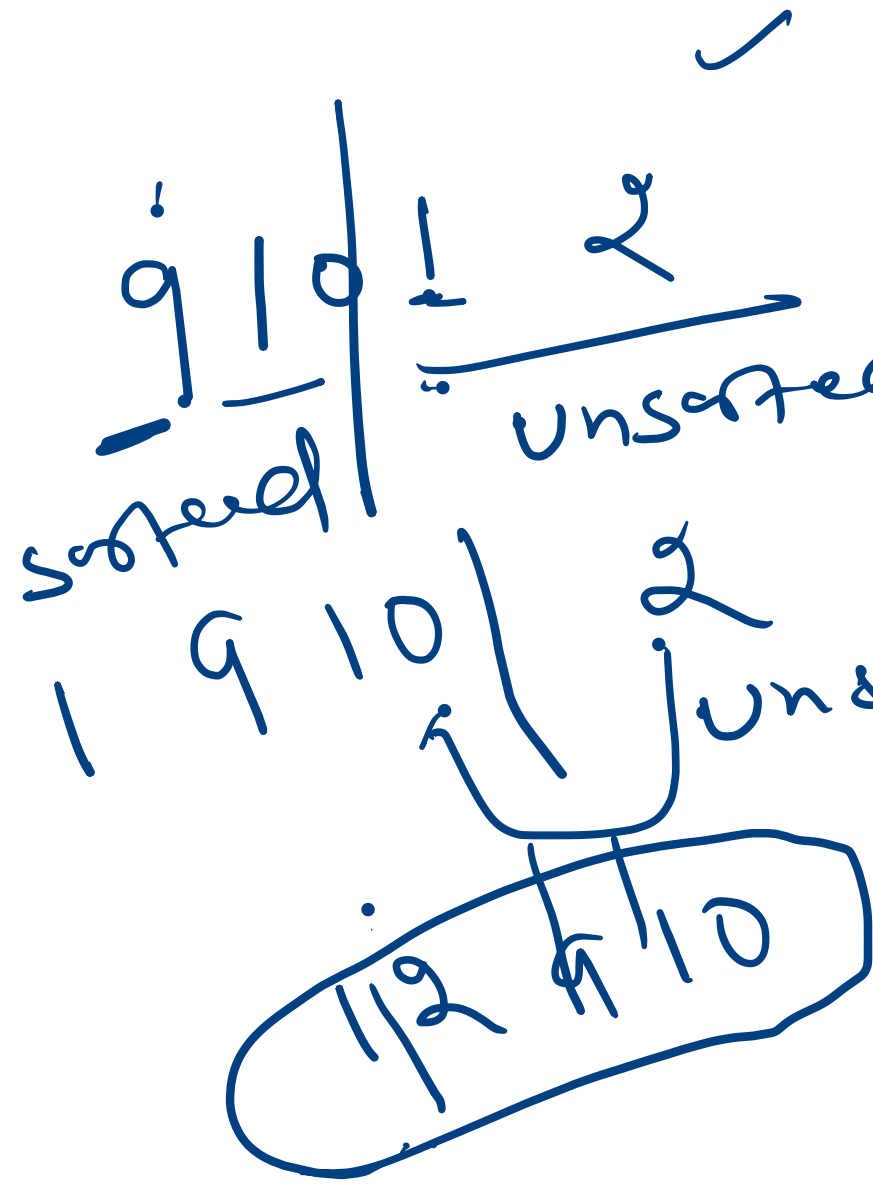

1 | 4 | 2 | 5 | 8
pass = 4
2 < 4
1 | 2 | 4 | 5 | 8
1 | 2 | 4 | 5 | 8
1 | 2 | 4 | 5 | 8
compare elements
swap
1, 2, 4, 5, 8
n = 5
pass = n - 1
= 5 - 1



} swapping elements

swapping set the elements


10
9
1
2
Card already sorted


9 10 1 2
sorted
unsorted
1 9 10 1 2
unsorted
12 4 10

| 5 | 1 | 3 | 4 | 6 | 2 |

selection sort.

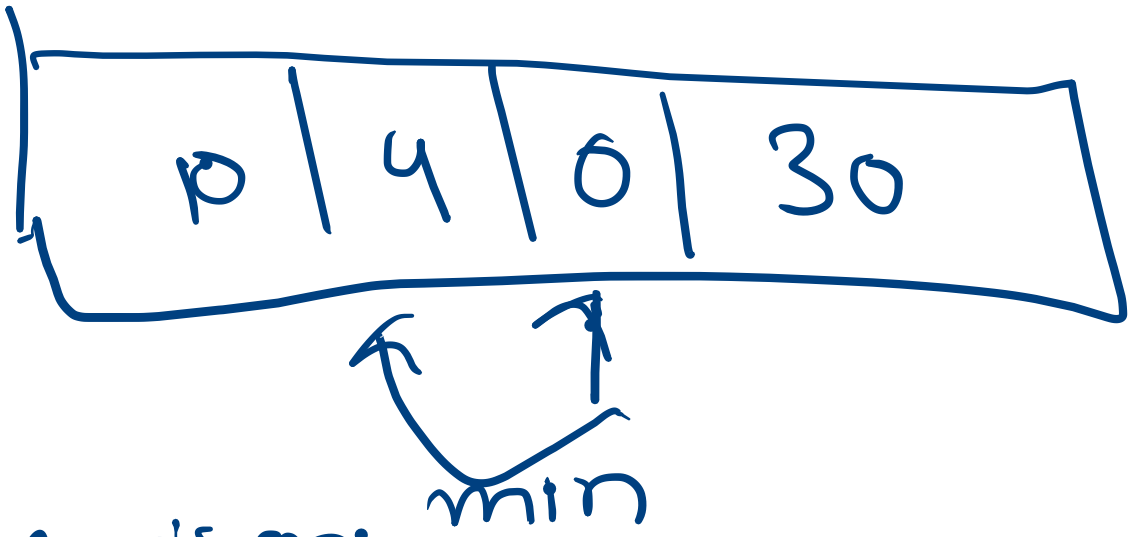
↳ comparison. (call elements and find out min. elem)

↳ swap min. with. first position.

selection. sort.

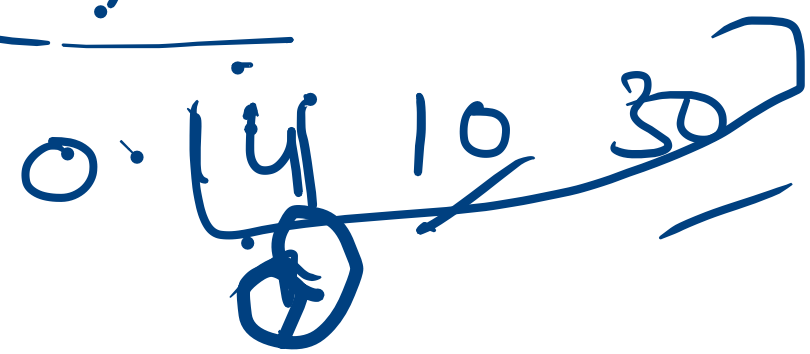
| 1 | 5 | 3 | 4 | 6 | 2. |

↑ sorted position..

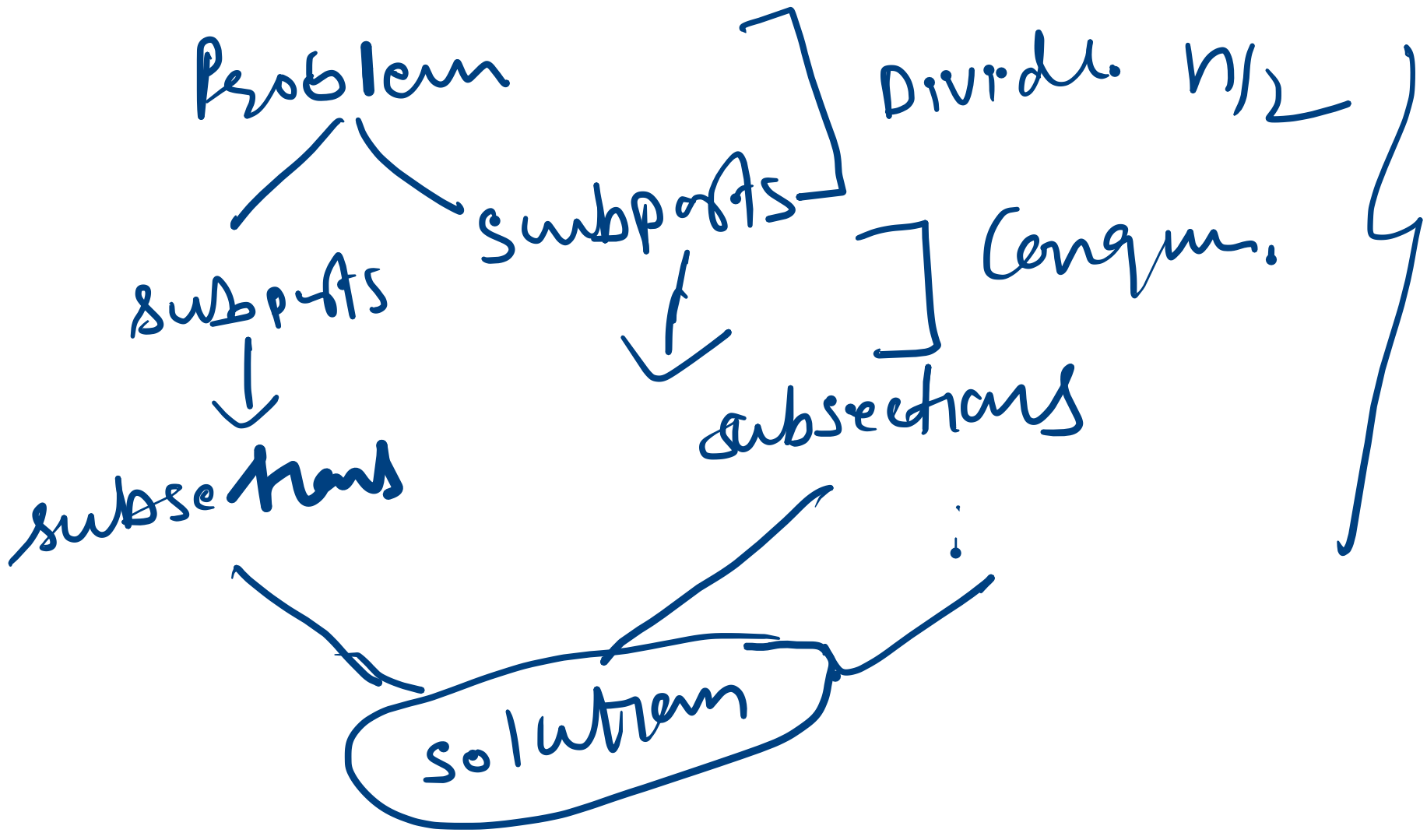


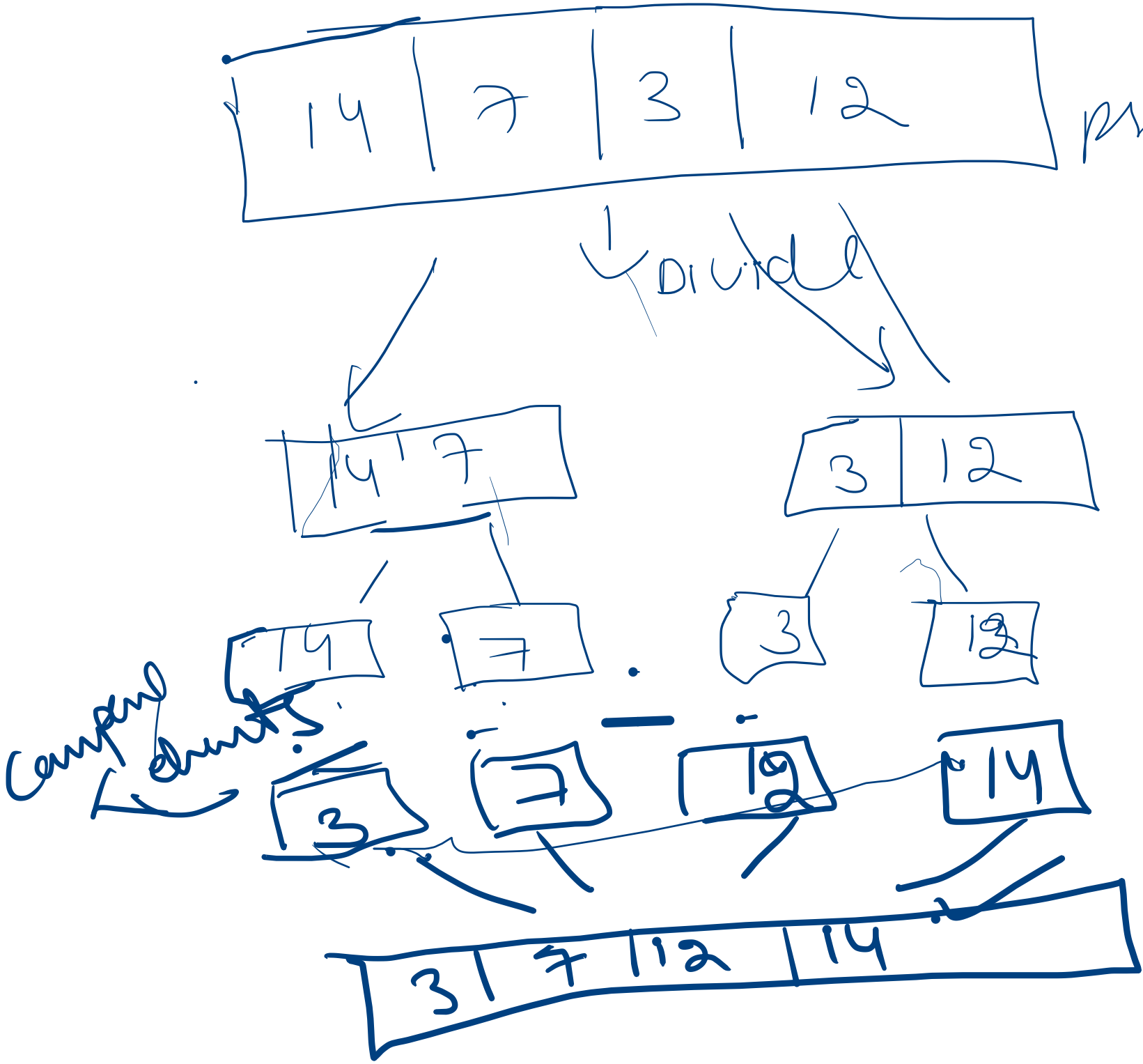
[Selection sort.]

1) Comparison.
2) find minimum. value.
3) [swap with. first index value.]



merge sort (Divide & conquer)




14 | 7 | 3 | 12
problem.
divide
merge sort.
14 | 7
3 | 12
14
7
3
12
compare units
3
7
12
14
3 7 12 14

Sorting algo.

→ Bubble sort ✓
→ selection sort ✓
→ Insertion sort. ✓
→ merge. sort ✓
→ quick sort ✓

8 | 2 4 9 3 6 (insertion sort)

sorted | unsorted.

2 < 8 | 4 9 3 6 unsorted.

2 8 | 4 9 3 6

sorted | unsorted.

2 4 8 | 9 3 6

sorted | unsorted.

2 4 8 9 | 3 6

sorted | unsorted

2 3 4 8 9 | 6

sorted | unsorted

2 3 4 6 8 9

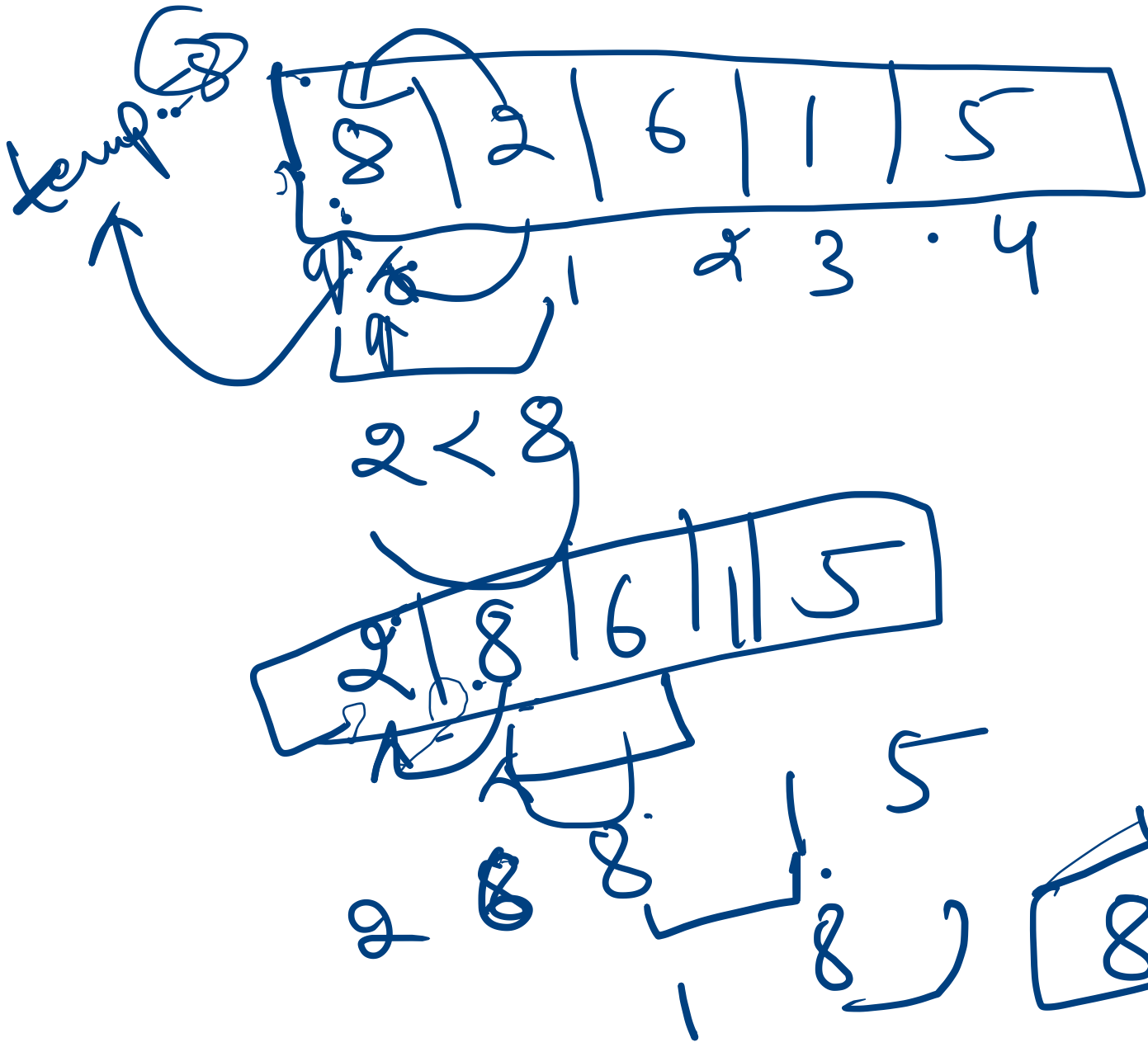
4 < 8 . 3 6 unsorted.

2 < 4 , 4 < 8 . 9 . 3 6

2 4 8 | 9 3 6

unsorted.

9 > 2, 4, 8



n = 5

passes = n - 1

= 4