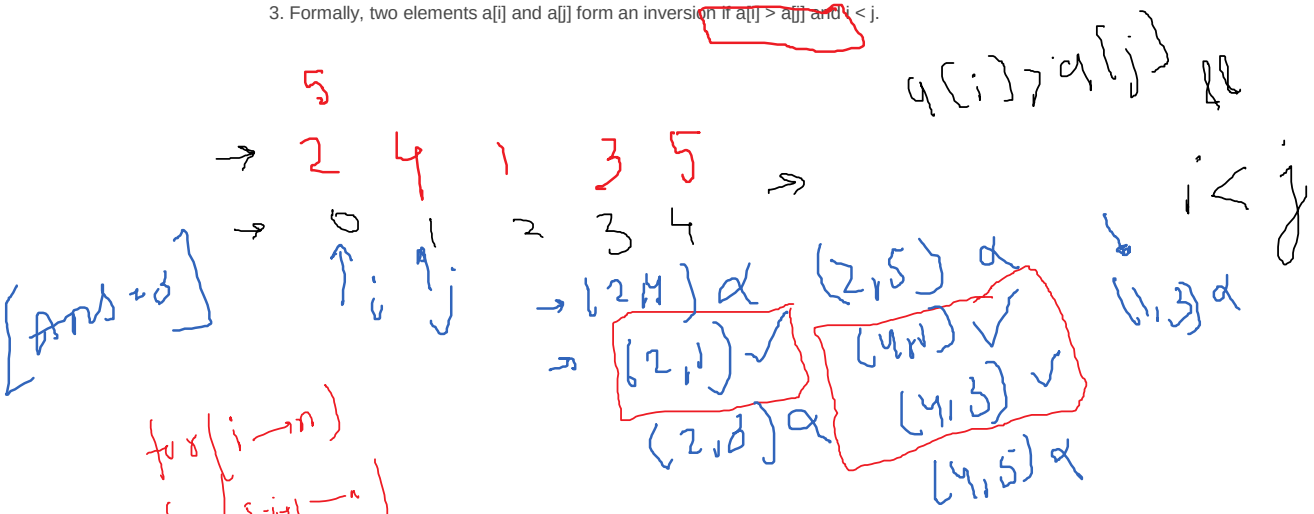
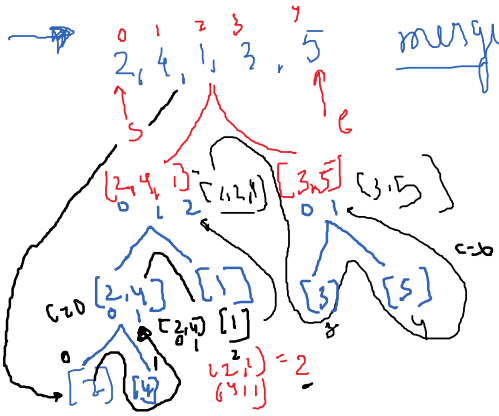
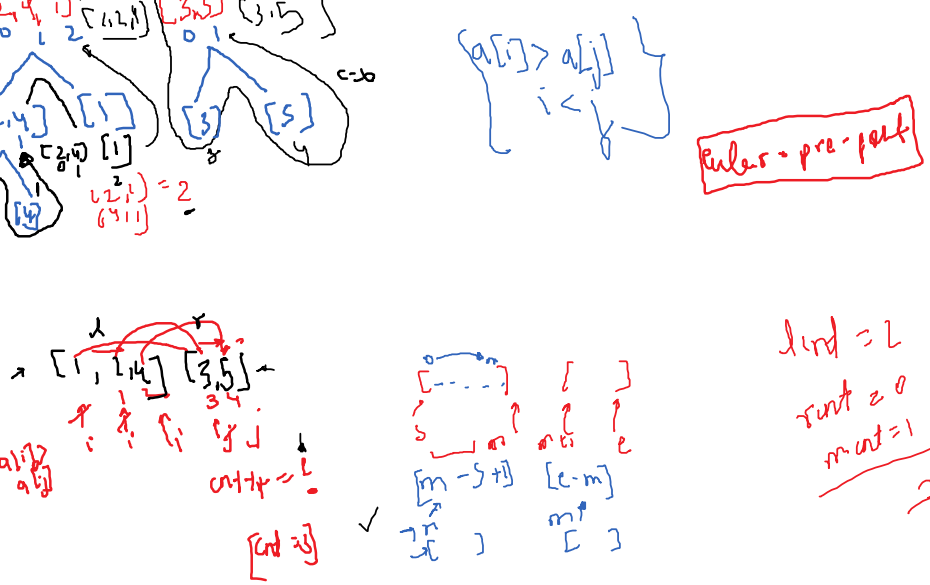
10 July 2022 21:30

1. Given an array of integers. Find the Inversion Count in the array.
2. For an array, inversion count indicates how far (or close) the array is from being sorted. If array is already sorted then the inversion count is 0. If an array is sorted in the reverse order then the inversion count is the maximum.
3. Formally, two elements a[i] and a[j] form an inversion if a[i] > a[j] and i < j.



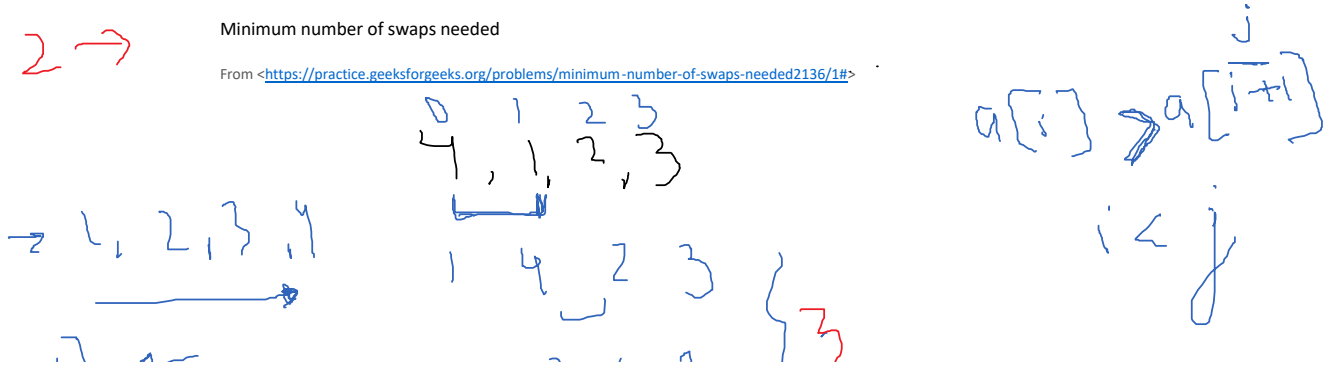


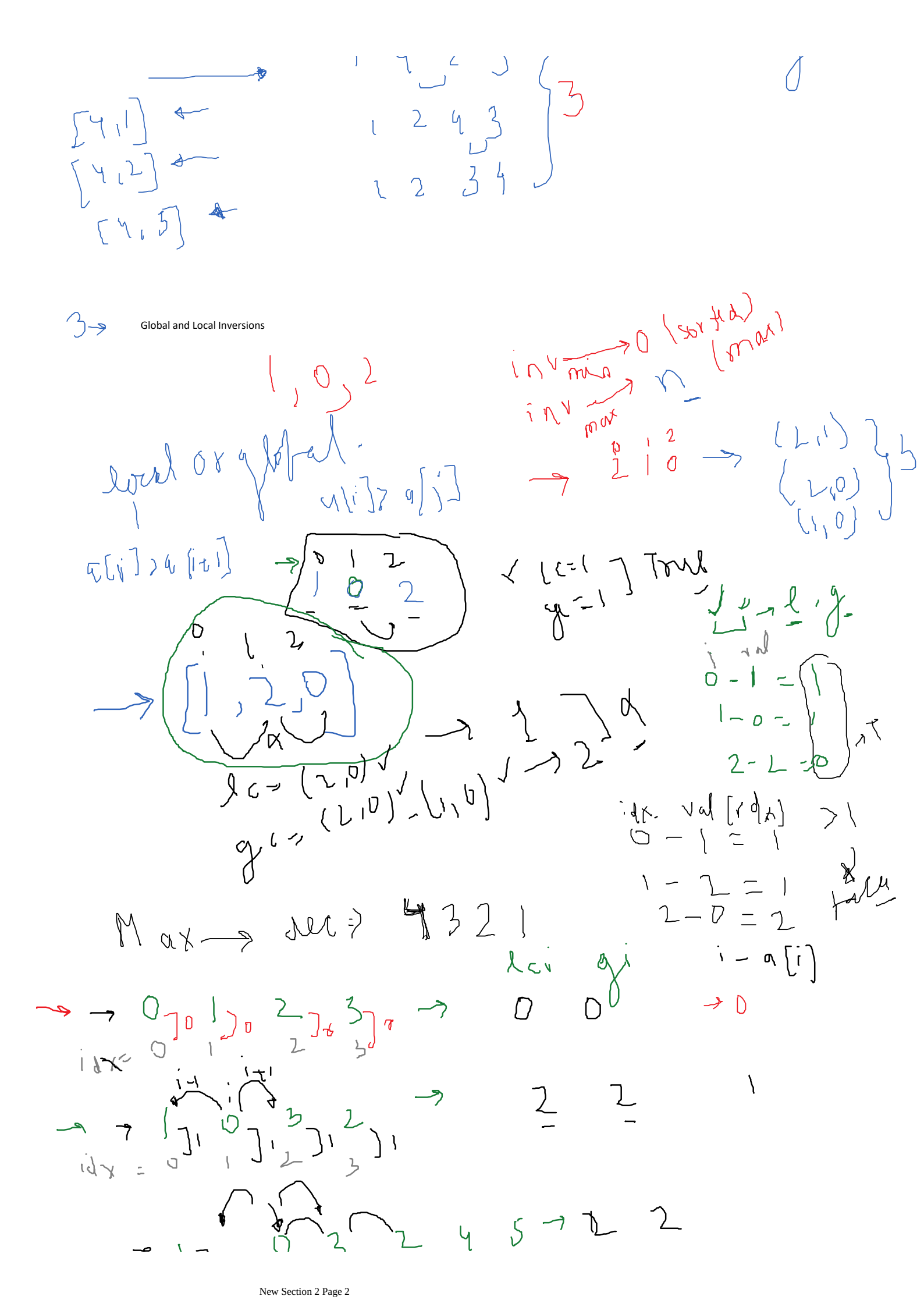
```
int cntinversion(vector<int>&arr , int n)
{
   int ivr = 0 ;
   for(int i = 0 ; i< n ; i++)
   {
      for(int j = i + 1 ; j < n ; j++)
      {
         if(arr[i] > arr[j] && i < j)
         {
            ivr++;
         }
      }
   }
   return ivr ;
}
```



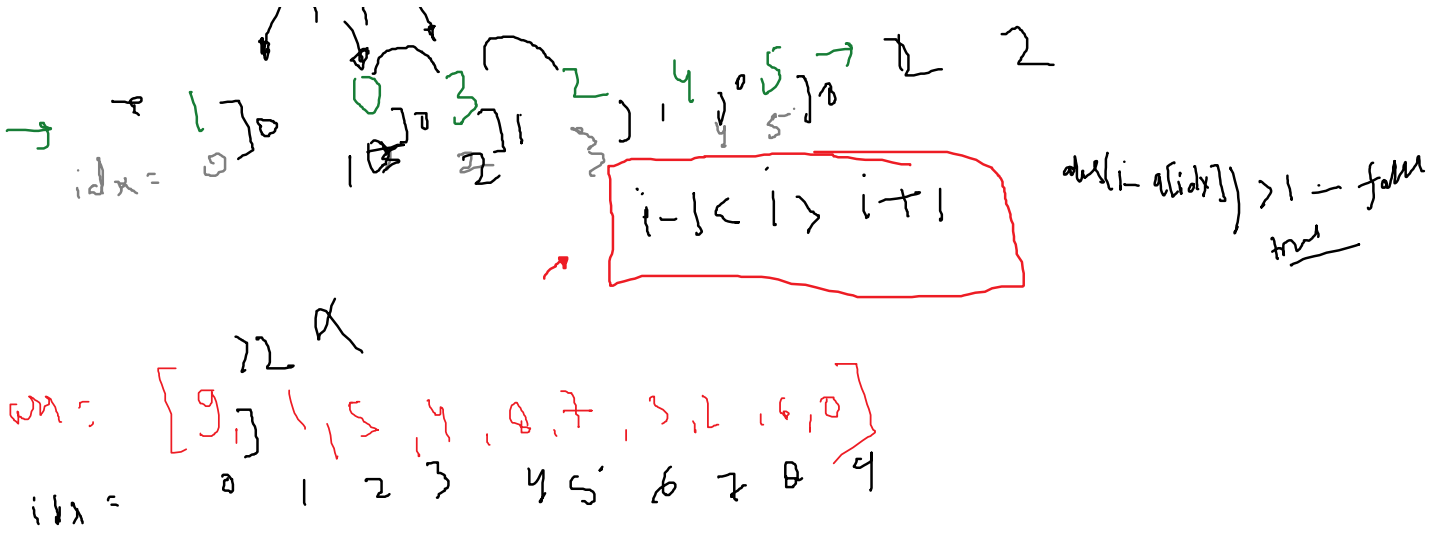
Minimum number of swaps needed

From <https://practice.geeksforgeeks.org/problems/minimum-number-of-swaps-needed2136/1#>



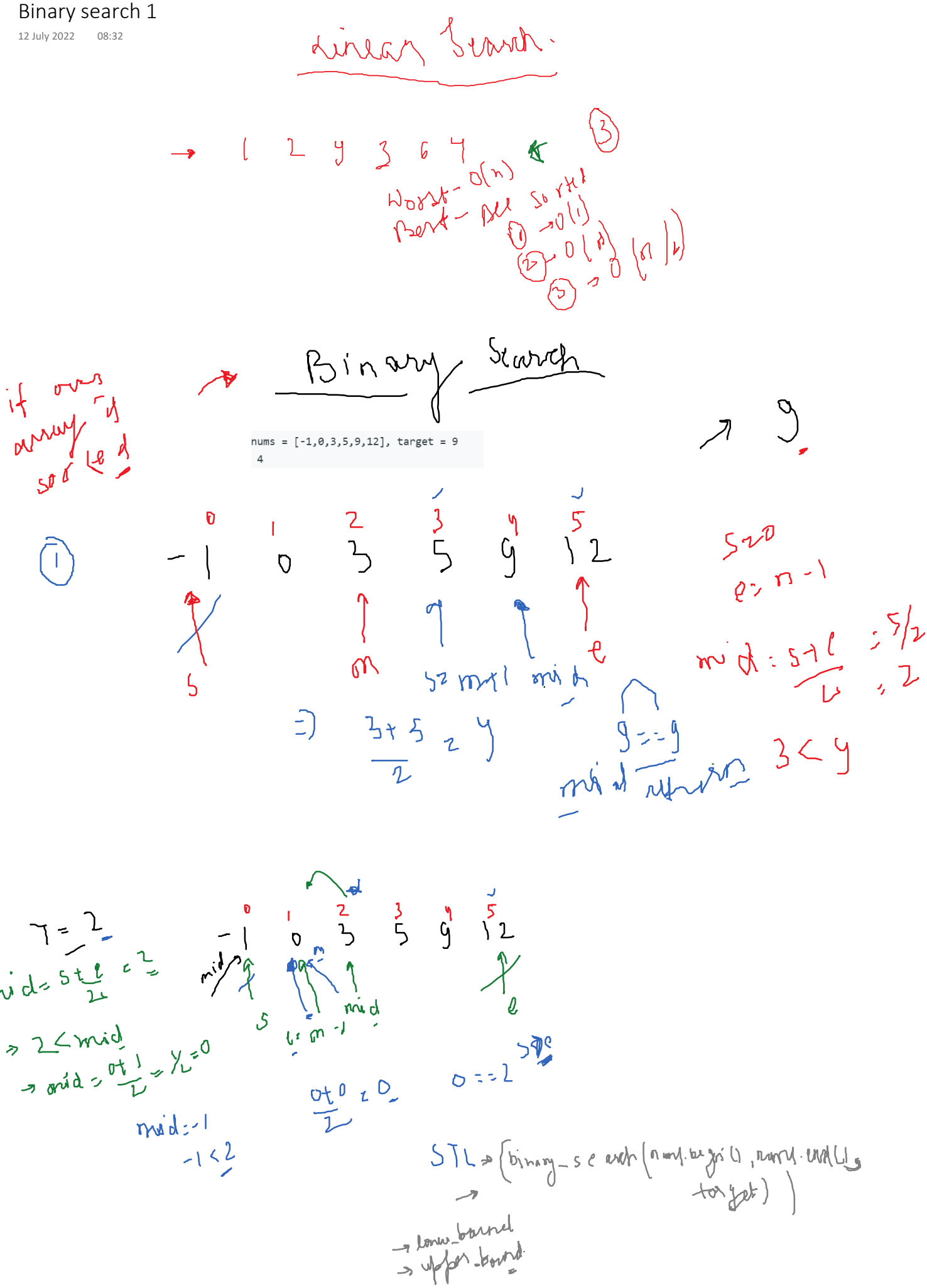


3→ Global and Local Inversions


idx =
i-1 < i > i+1
abs(i - q[idx]) > 1 — false
true
arr:
idx =

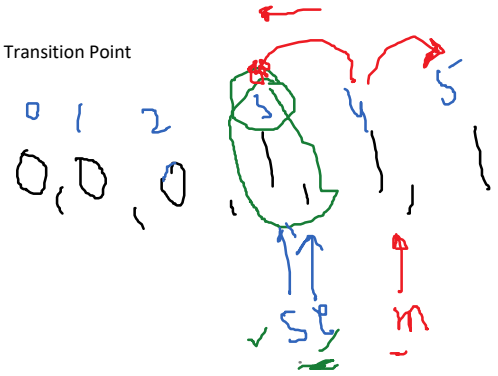
Binary search 1

12 July 2022 08:32

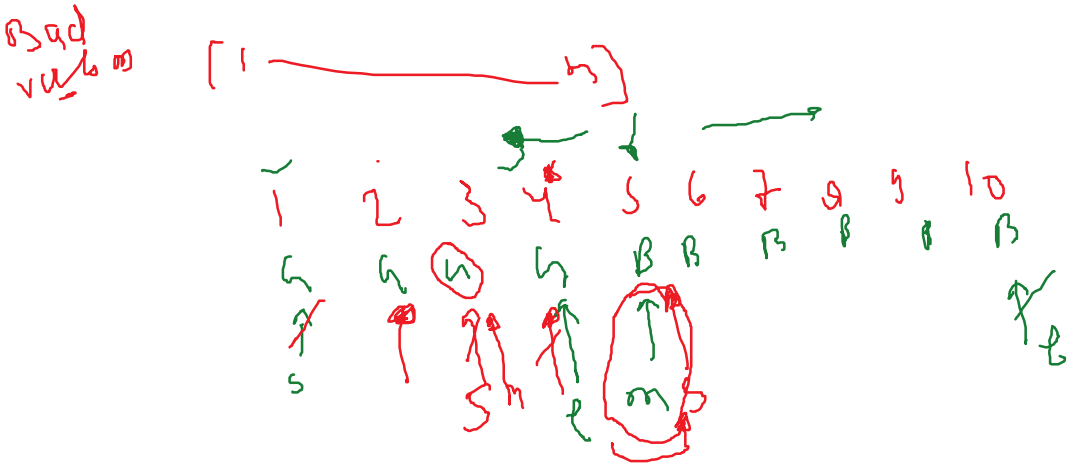


3

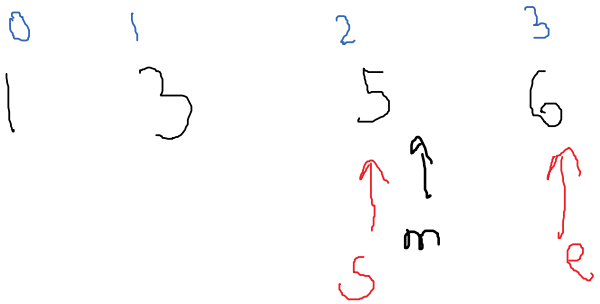
Find Transition Point



3

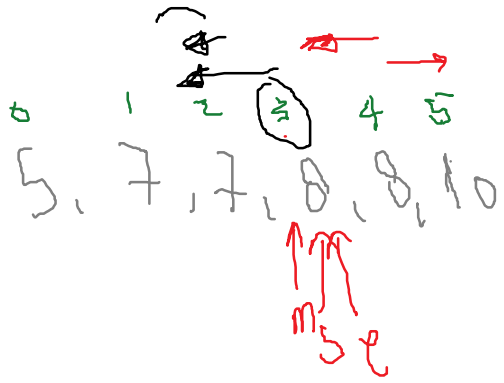


4 — Search Insert Position



Find First and Last Position of Element in Sorted Array

① first idx
② last idx



t: 0
ans: -1
ans: mid
3

fidx

Binary Search 2

12 July 2022 22:56

No of occ

```
N = 7, X = 2
Arr[] = {1, 1, 2, 2, 2, 2, 3}
Output: 4
```

[1, 1, 2, 2, 2, 2, 3]

5 - 2 + 1 = 4

Broken & wrong

f = mid

[5, 10, 15, 22, 33, 40, 42, 55]

22 < 25

33 < 25

1. val > mid
2. val < mid

[7, 14, 18, 25, 30]

[5, 6, 9, 10, 11]

f = -1, c = 5

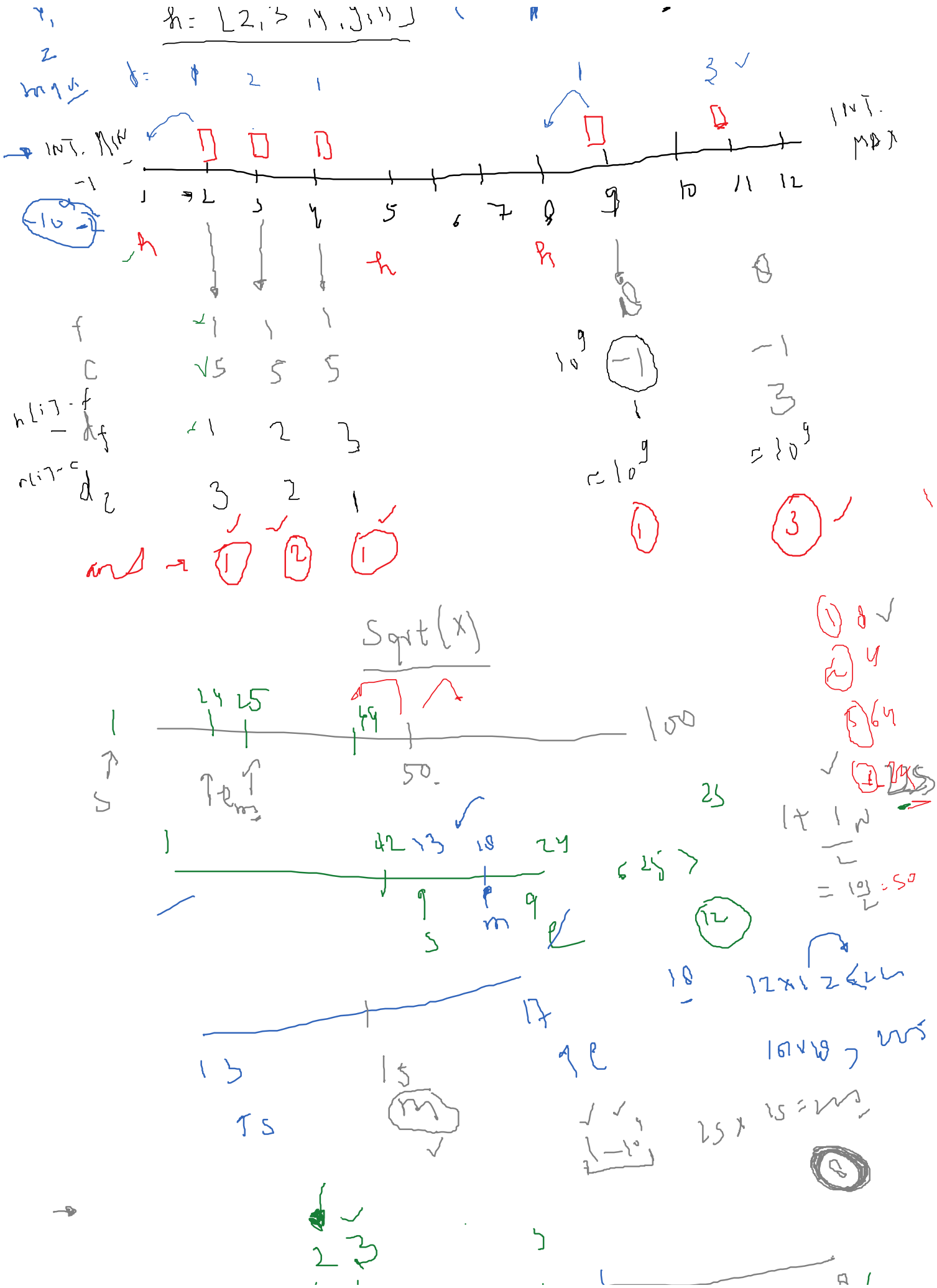
f = 11, c = -1

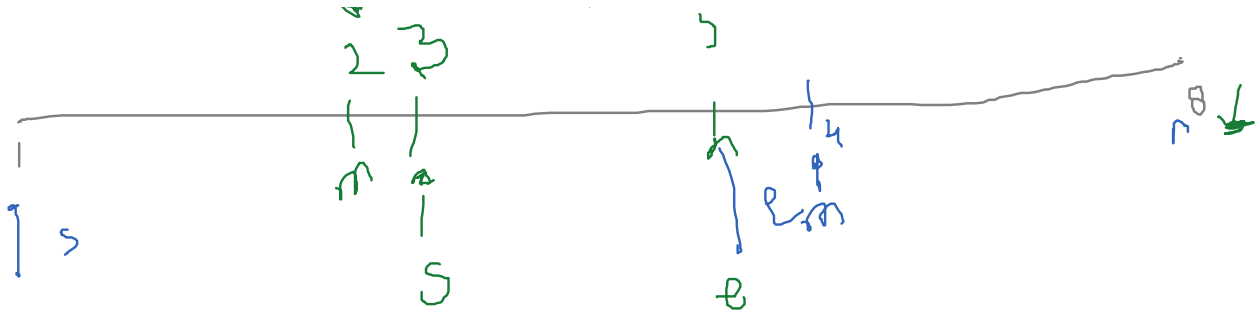
f = 5, c = 5

3 → Heaters

h = [2, 3, 4, 9, 11]

heaters = [1, 5, 8]


Sqrt(x)



Binary search 3

14 July 2022 08:43

Find Minimum in Rotated Sorted Array

Suppose an array of length `n` sorted in ascending order is **rotated** between `1` and `n` times. For example, the array `nums = [0,1,2,4,5,6,7]` might become:

- `[4,5,6,7,0,1,2]` if it was rotated `4` times.
- `[0,1,2,4,5,6,7]` if it was rotated `7` times.

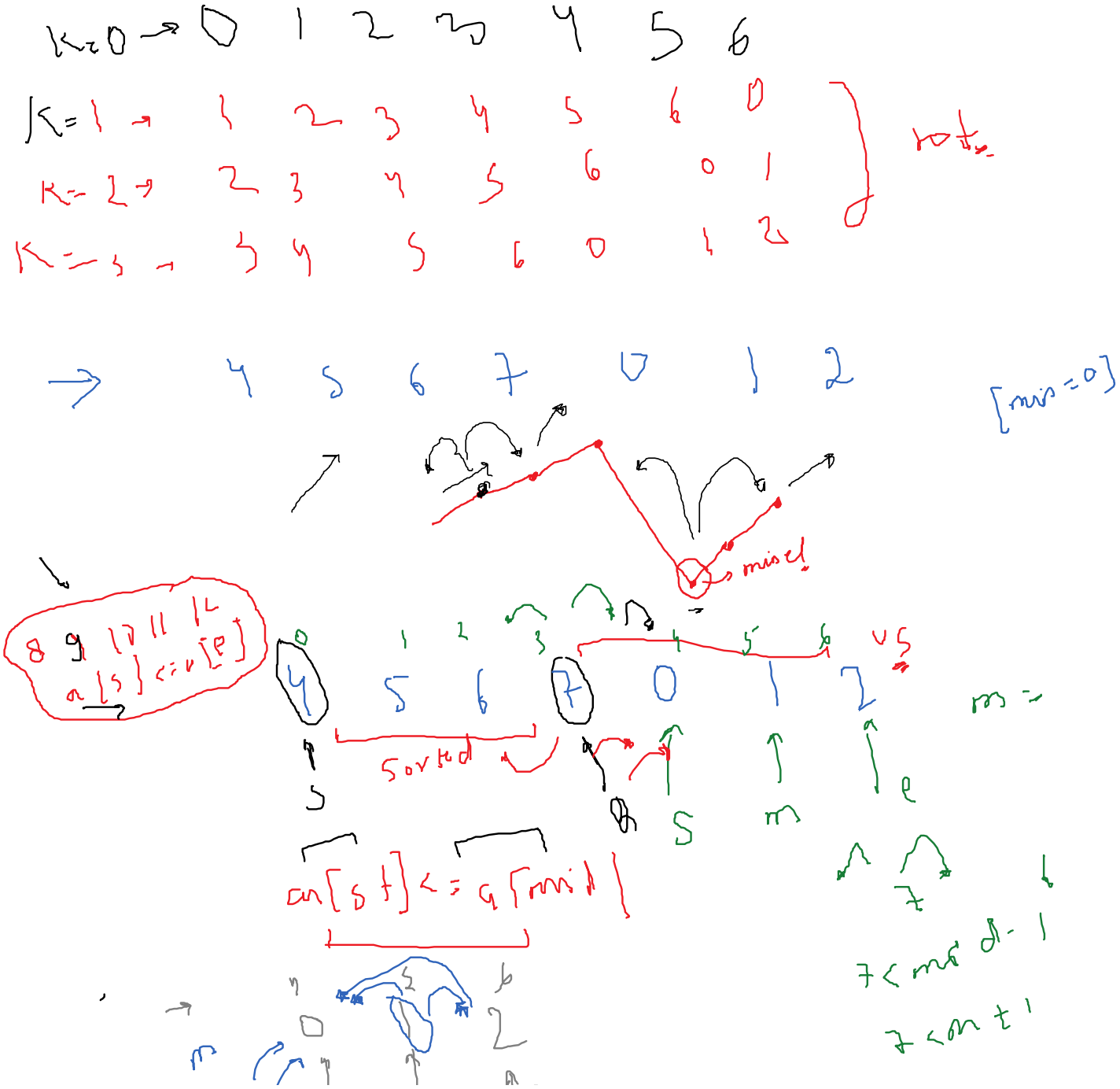
Notice that **rotating** an array `[a[0], a[1], a[2], ..., a[n-1]]` 1 time results in the array `[a[n-1], a[0], a[1], a[2], ..., a[n-2]]`.

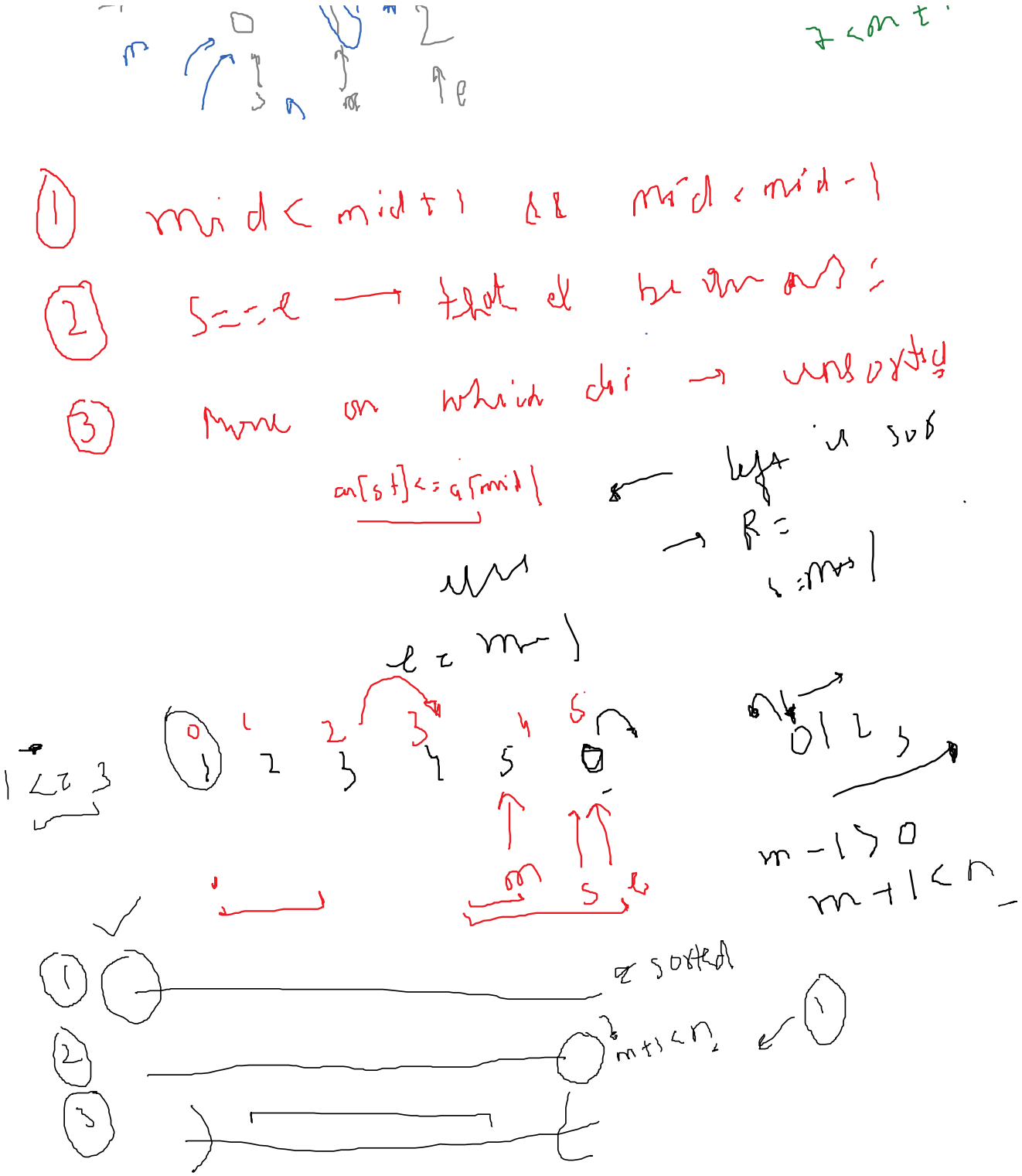
Given the sorted rotated array `nums` of **unique** elements, return *the minimum element of this array*.

You must write an algorithm that runs in `O(log n) time`.

```
[4,5,6,7,0,1,2]
```

Screen clipping taken: 14-07-2022 08:46




Number of Rotations
n = 3

Search a 2D Matrix

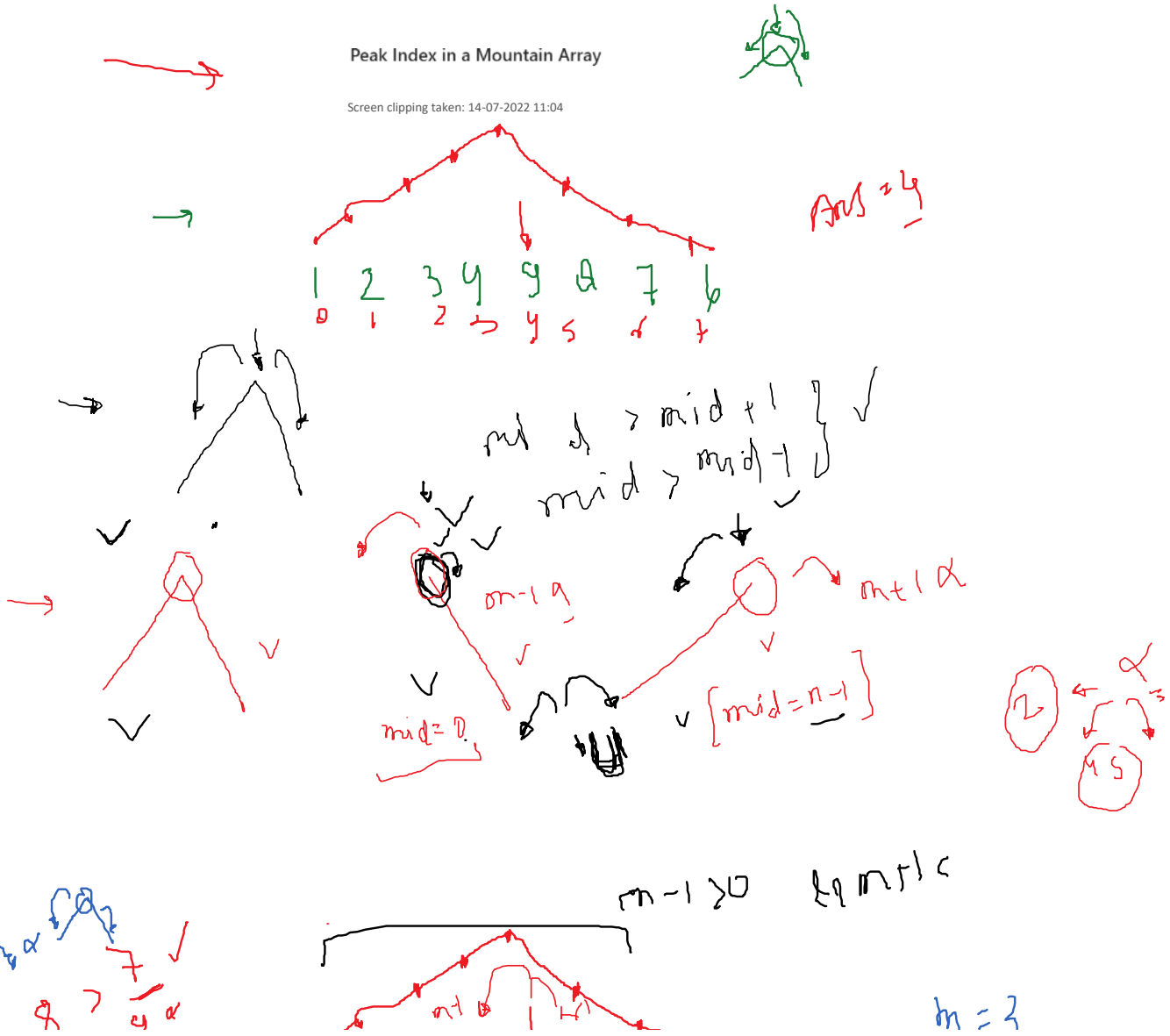
1	3	5	7
10	11	16	20
23	30	34	60

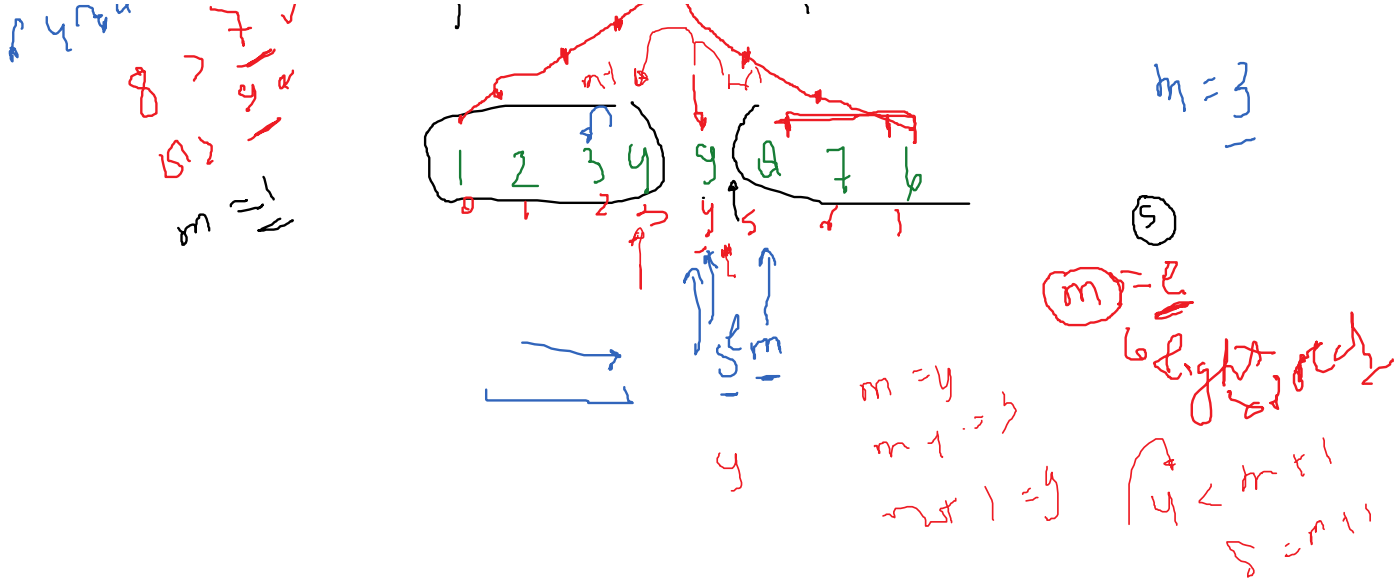
```
Input: matrix = [[1,3,5,7],[10,11,16,20],[23,30,34,60]], target = 3
Output: true
```

Screen clipping taken: 14-07-2022 10:20

Peak Index in a Mountain Array

Screen clipping taken: 14-07-2022 11:04



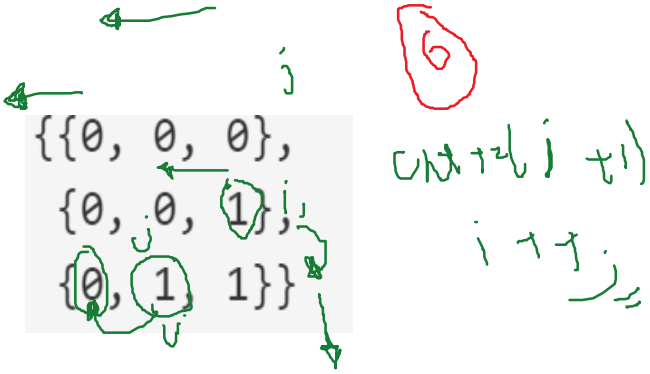


Binary search 4

15 July 2022 08:47

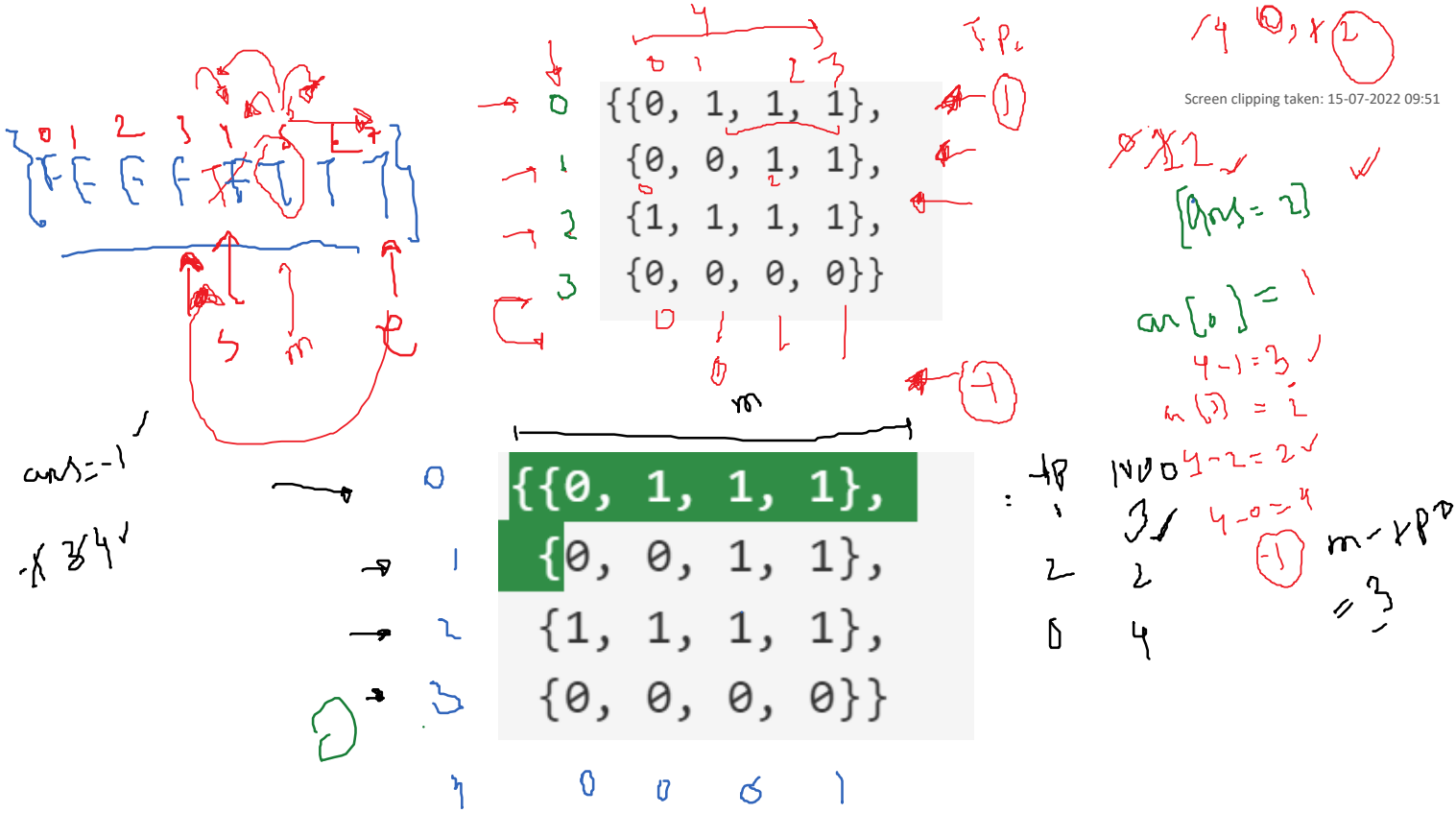
Count zeros in a sorted matrix

```
N = 3
A = {{0, 0, 0},
     {0, 0, 1},
     {0, 1, 1}}
Output: 6
Explanation:
The first, second and third row contains 3, 2
zeroes respectively.
```



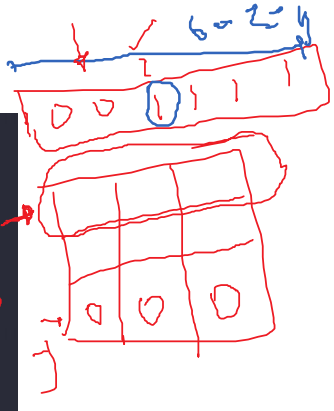
Screen clipping taken: 15-07-2022 08:49

Row with max 1

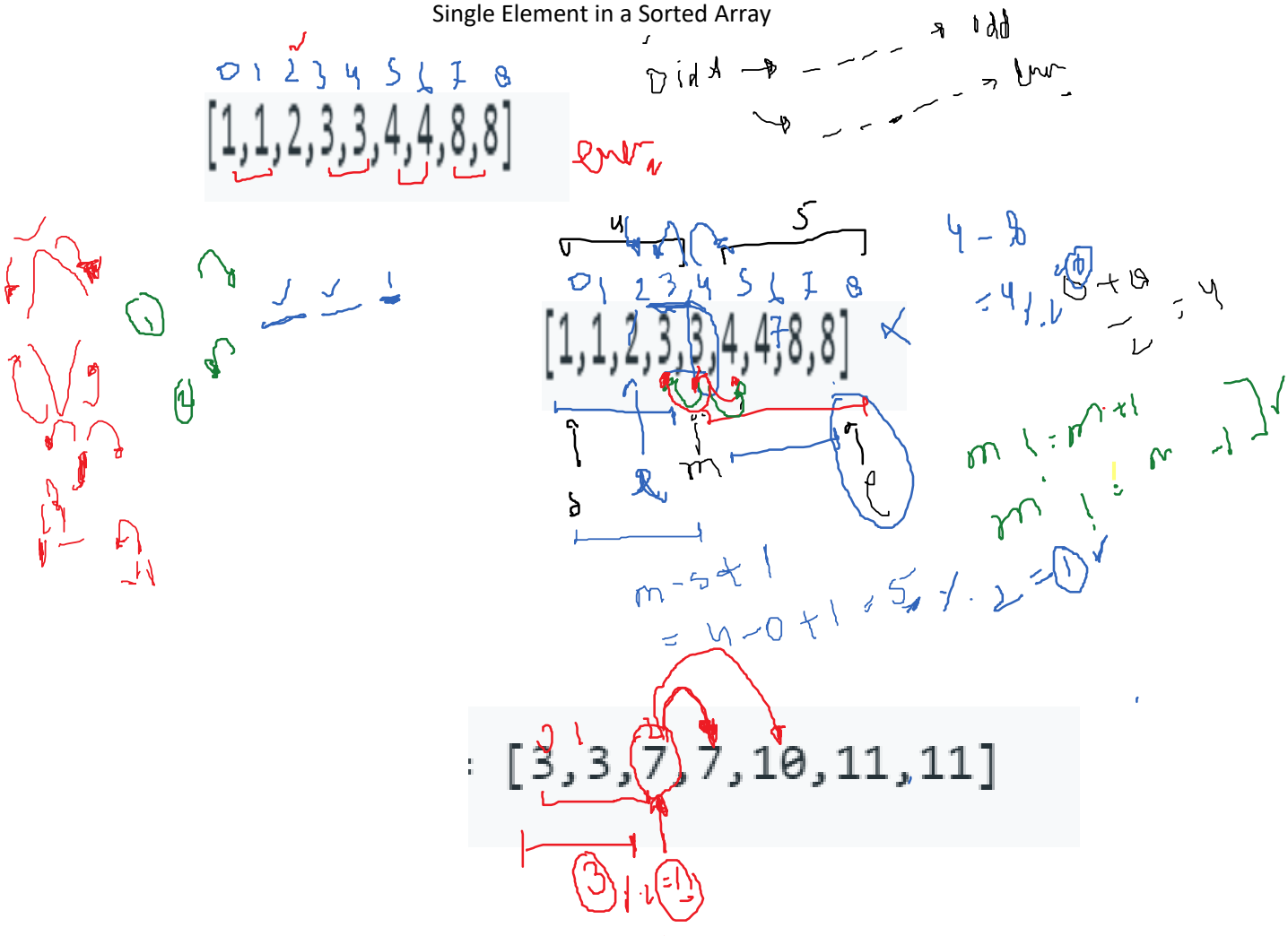


Screen clipping taken: 15-07-2022 09:51

```cpp
int rowWithMax1s(vector<vector<int> > arr, int n, int m) {
    int ans = -1 ;
    int noofones = 0;
    for(int i = 0 ; i< n; i++)
    {
        int tpo = tp(arr[i], 0 , m - 1) ;
        if(tpo != -1 && m - tpo > noofones)
        {
            ans = i;
            noofones = m - tpo ;
        }
    }
    return ans ;
}
```



Single Element in a Sorted Array



Binary search on ans

16 July 2022 09:02

Allocate minimum number of pages

You are given **N** number of books. Every **ith** book has **Ai** number of pages. The books are arranged in ascending order.

You have to allocate contiguous books to **M** number of students. There can be many ways or permutations to do so. In each permutation, one of the M students will be allocated the maximum number of pages. Out of all these permutations, the task is to find that particular permutation in which the maximum number of pages allocated to a student is the minimum of those in all the other permutations and print this minimum value.

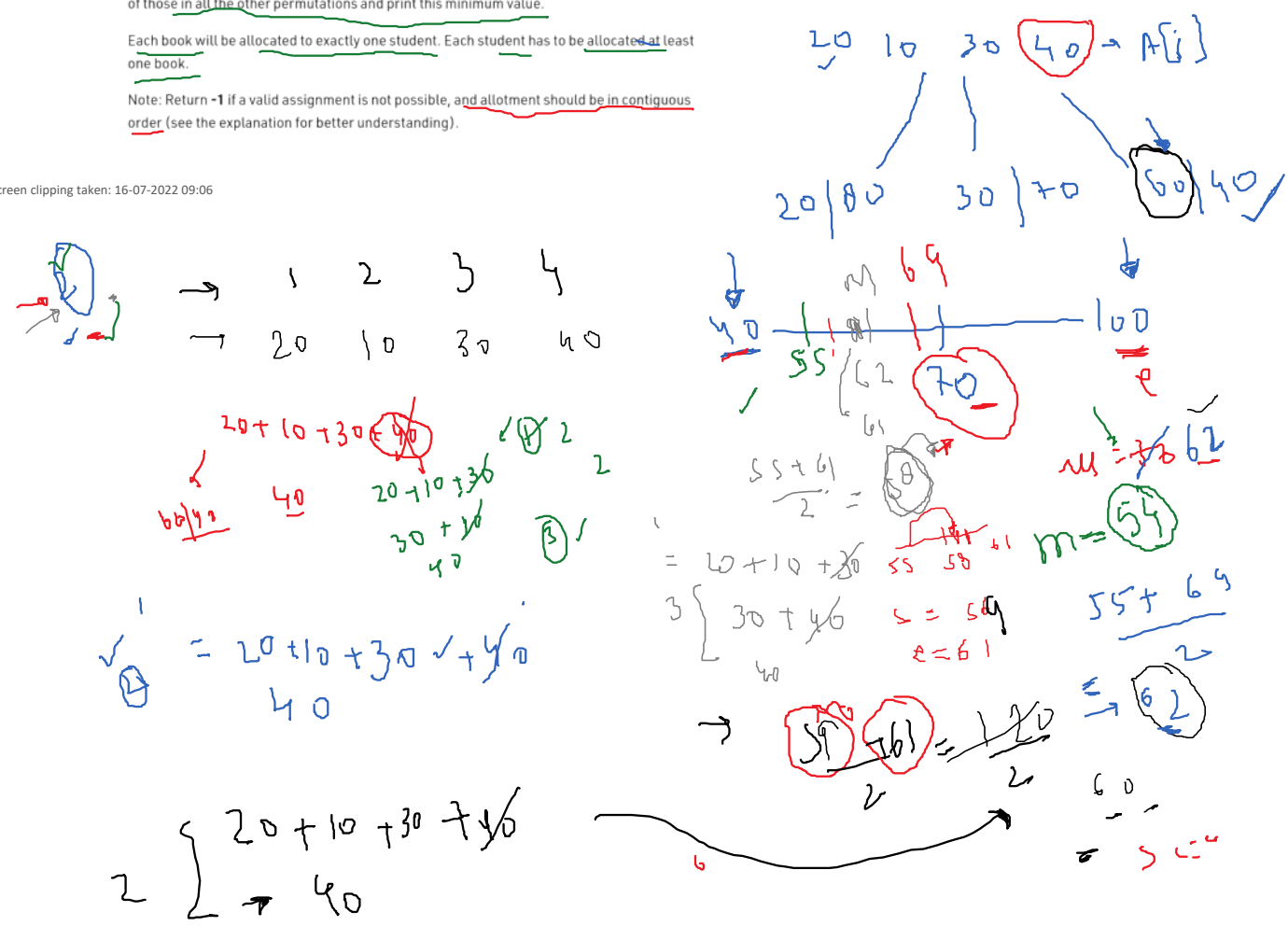
Each book will be allocated to exactly one student. Each student has to be allocated at least one book.

Note: Return **-1** if a valid assignment is not possible, and allotment should be in contiguous order (see the explanation for better understanding).

N = 4
A[] = {12,34,67,90}
M = 2
Output:113

Screen clipping taken: 16-07-2022 09:08

Screen clipping taken: 16-07-2022 09:06



Painter's Partition Problem

Given 2 integers **A** and **B** and an array of integars **C** of size **N**.

Element **C[i]** represents length of **ith** board.

You have to paint all **N** boards **[C0, C1, C2, C3 ... CN-1]**. There are **A** painters available and each of them takes **B** units of time to paint **1 unit** of board.

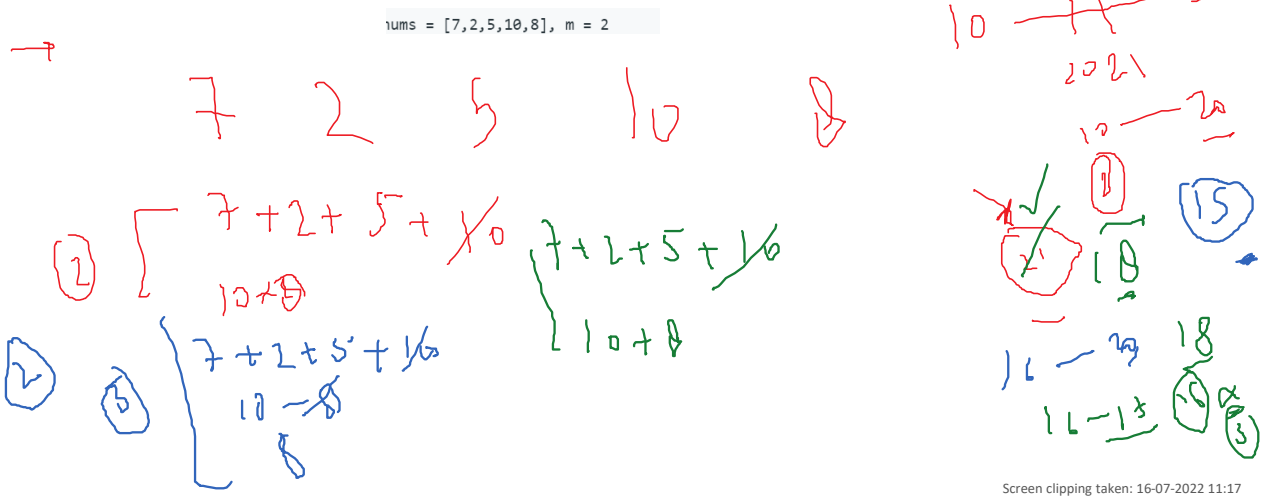
Calculate and return minimum time required to paint all boards under the constraints that **any painter will only paint contiguous sections of board.**

- 2 painters cannot share a board to paint. That is to say, a board
cannot be painted partially by one painter, and
partially by another.
- A painter will only paint contiguous boards. Which means a
configuration where painter 1 paints board 1 and 3
but not 2 is
invalid.

e ans % 10000003

- 2 painters cannot share a board to paint. That is to say, a board
cannot be painted partially by one painter, and

```
A = 2
B = 5
C = [1, 10]
```



Screen clipping taken: 16-07-2022 11:17

875. Koko Eating Bananas

Medium 4566 202 Add to List Share

Koko loves to eat bananas. There are `n` piles of bananas, the `i`th pile has `piles[i]` bananas. The guards have gone and will come back in `h` hours.

Koko can decide her bananas-per-hour eating speed of `k`. Each hour, she chooses some pile of bananas and eats `k` bananas from that pile. If the pile has less than `k` bananas, she eats all of them instead and will not eat any more bananas during this hour.

Koko likes to eat slowly but still wants to finish eating all the bananas before the guards return.

Return *the minimum integer* `k` *such that she can eat all the bananas within* `h` *hours.*

Input: piles = [3,6,7,11], h = 8
Output: 4

xample 2:

Input: piles = [30,11,23,4,20], h = 5
Output: 30