Saturday, 24 July 2021 4:08 PM

Arrays

→ non-primitive / derived data type.

50 students
→ 50 variables → 50 placeholders in memory

RAM

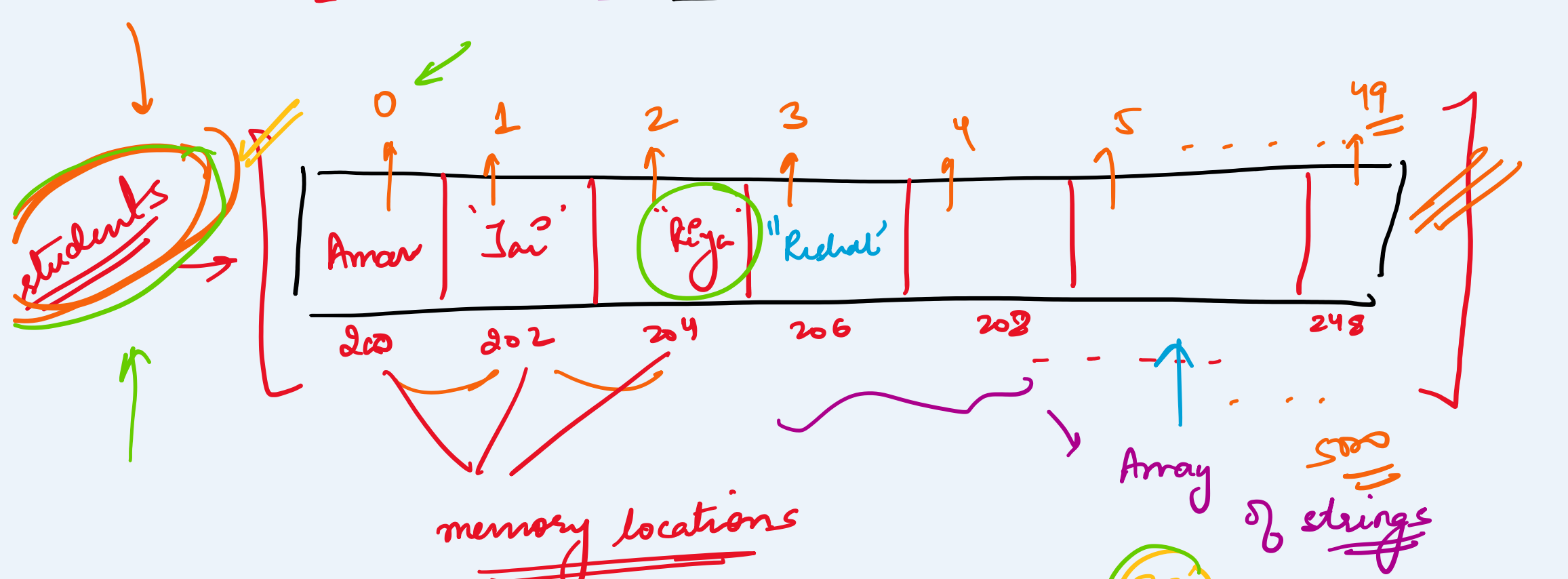
String name1 = "ganesh"
string name2 = "unnati"
⋮
name 50 = "__";

→ keep track of 50 variables

① Homogeneous

30,000 students → 30 variable names

[→ homogeneous data
→ contiguous memory location]



students [31]

indexing

students [3] = "Rishab"

→ whenever we are declaring an array, we need to declare its size in the start itself.

Arrays / 1-D Array

x-axis

0 1 2 3 . . . . . .

Array → fixed size
List → dynamic size

Syntax

int [ ] arr = new int [size];

int [ ] → data type
arr → name of array / var-name
int → datatype

arr

[I@6996db8 → address of memory location
[I → array of integers

arr is pointing to an array

Q. Given an array, print the max element of the array.

32, 43, 6, 91, 7, 22

output → 91

max

-2 -3 -4

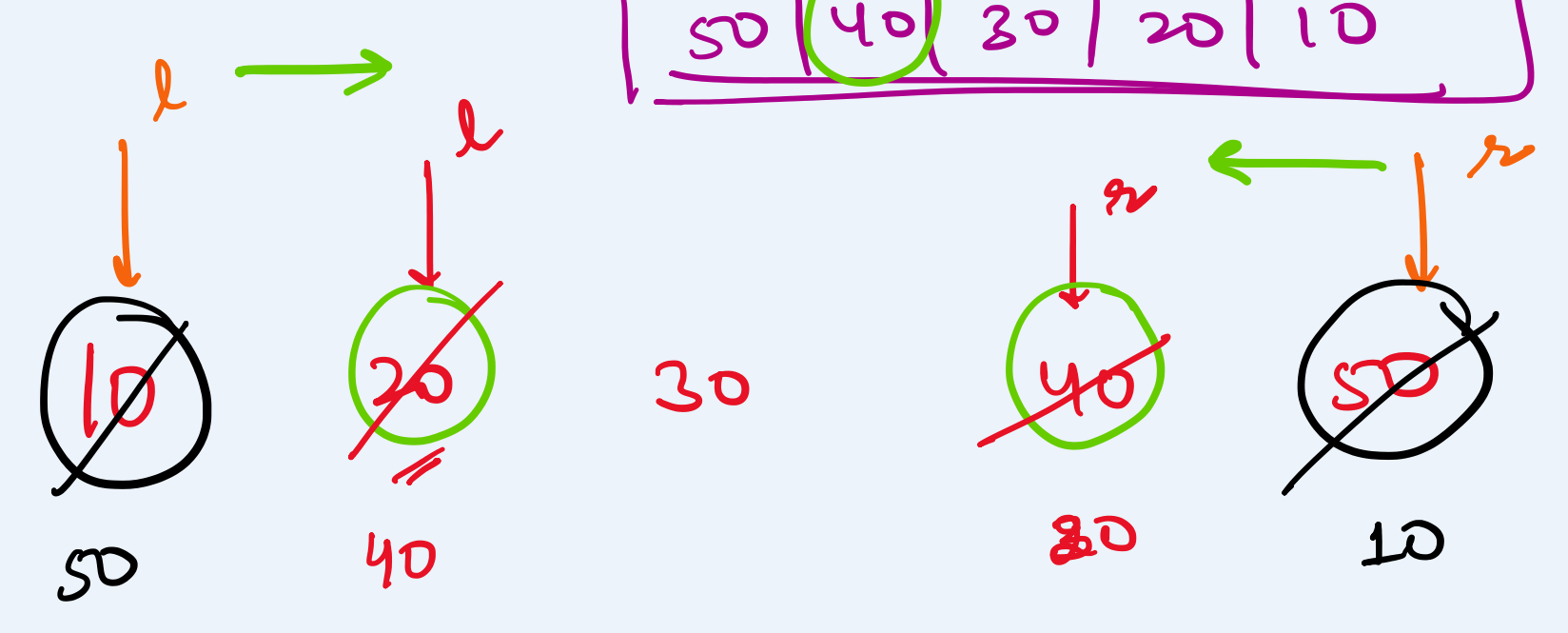
max = arr[0] / -∞

```
for (int i=0; i< arr.length; i++) {
    if (arr[i] > max)
        max = arr[i];
}
```

Q. Reverse the array

Eg: | 10 | 20 | 30 | 40 | 50 |

| 50 | 40 | 30 | 20 | 10 |




```
while ( left < right) {
    swap ( arr[left], arr[right]);
}
```

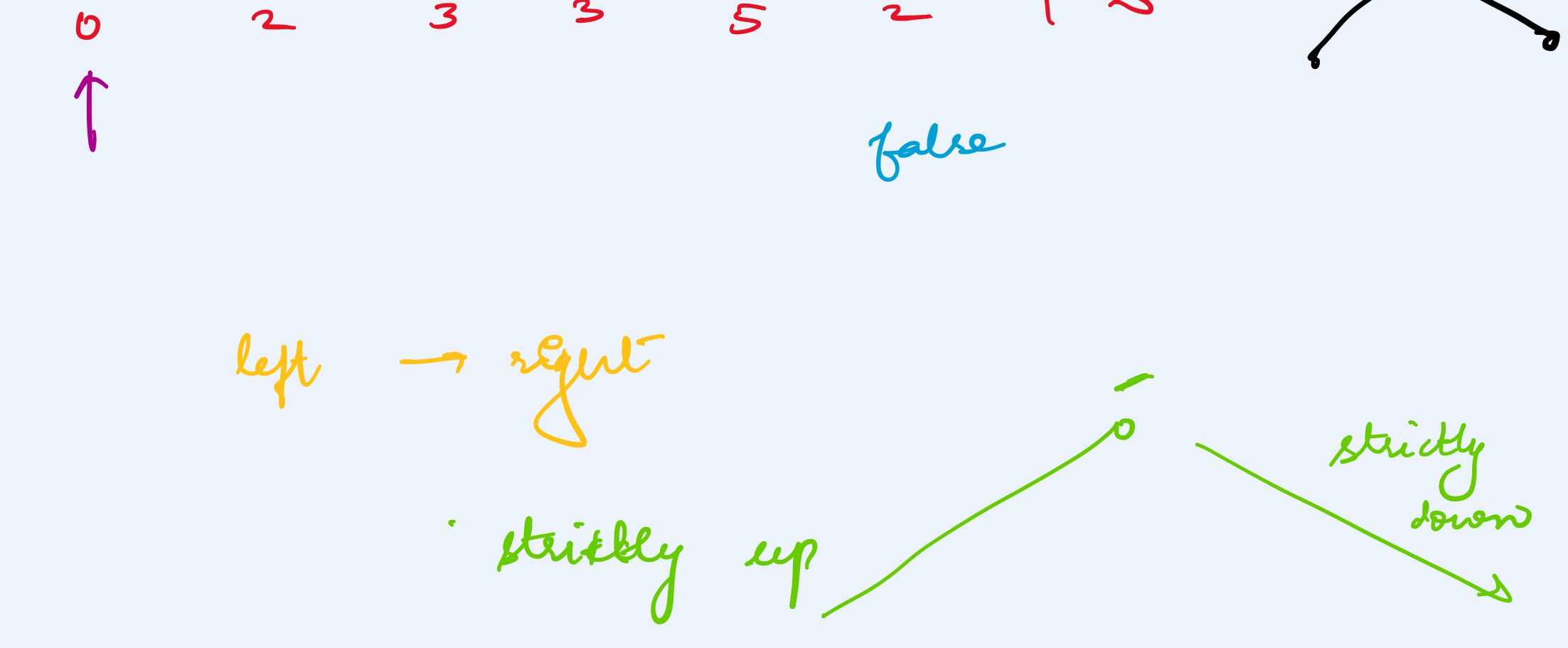
Q. Valid mountain array

Given an array of integers, return true, if and only if it is a valid mountain array.

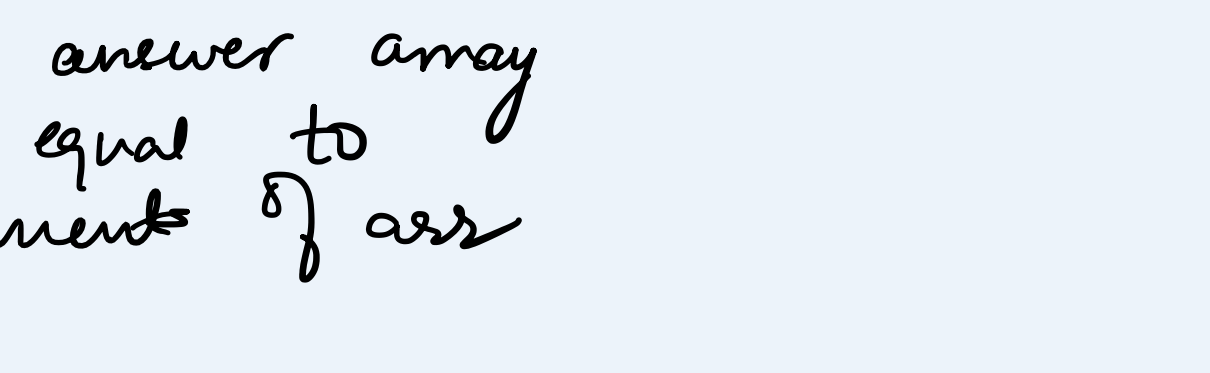
Arr is a mountain array if :-

① arr.length >= 3

② There exists some i for 0 < i < arr.length -1 such that
(a) arr[0] + arr[1] + arr[2] - - .. arr[i-1] + arr[i]
(b) arr [i] + arr [i+1] + - . . . arr [n-1]



left → right



Q. Given an array, return the answer array such that ans[i] is equal to the product of all elements of arr except arr[i].

Eg: arr = [1, 2, 3, 4]
output = [24, 12, 8, 6]

24 → 2×3×4
12 → 1×3×4
8 → 1×2×4
6 → 1×2×3

→ % operator ×