DAAPY Class 2

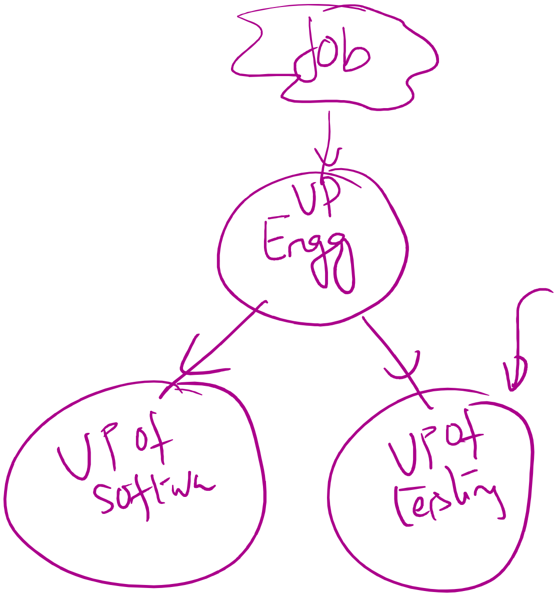
Wednesday, September 13, 2023 4:23 PM

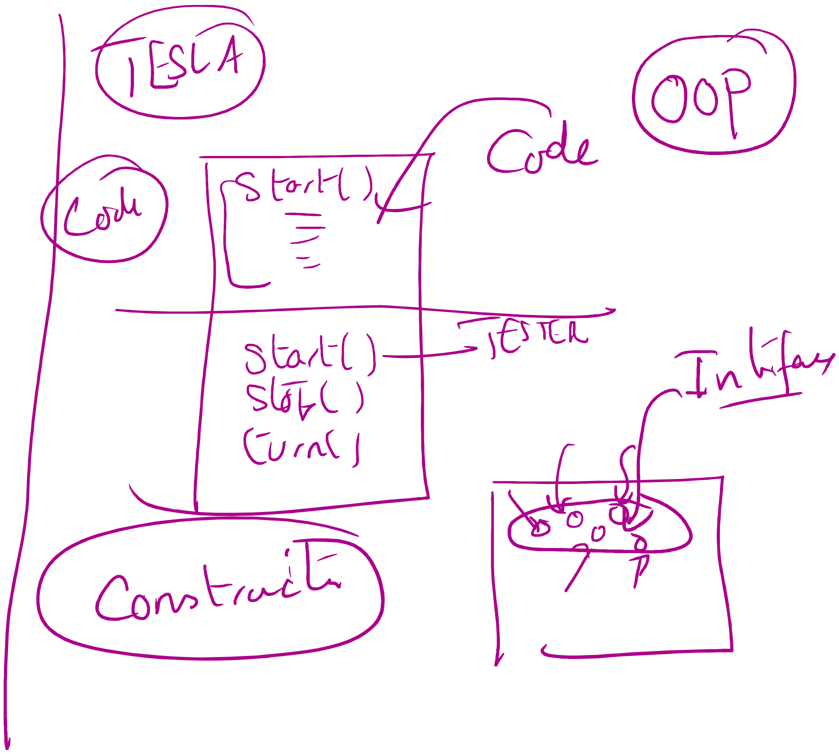
1. Recording
2. Screen
3. TALK

408.834.0284

1

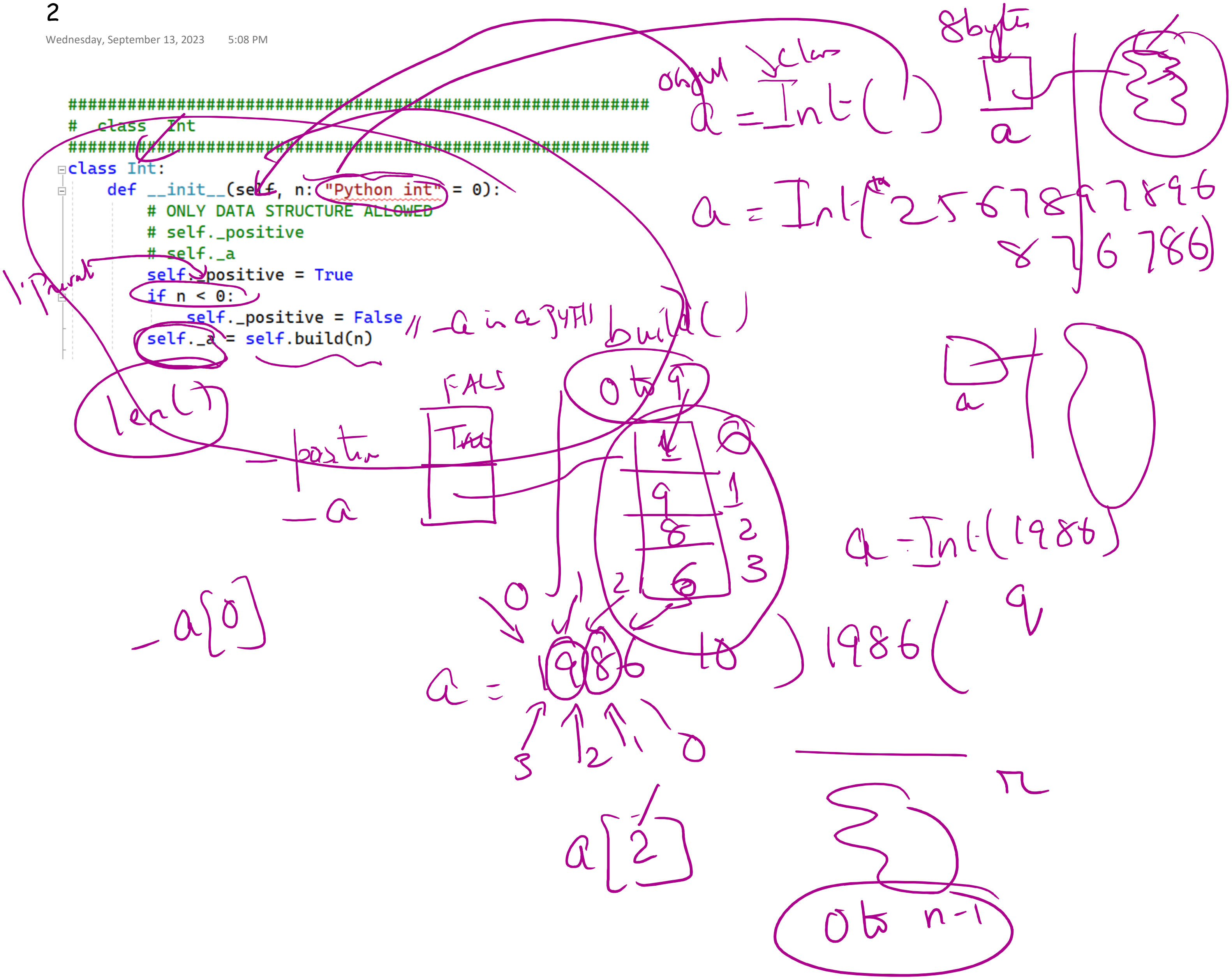
Wednesday, September 13, 2023 5:02 PM





2

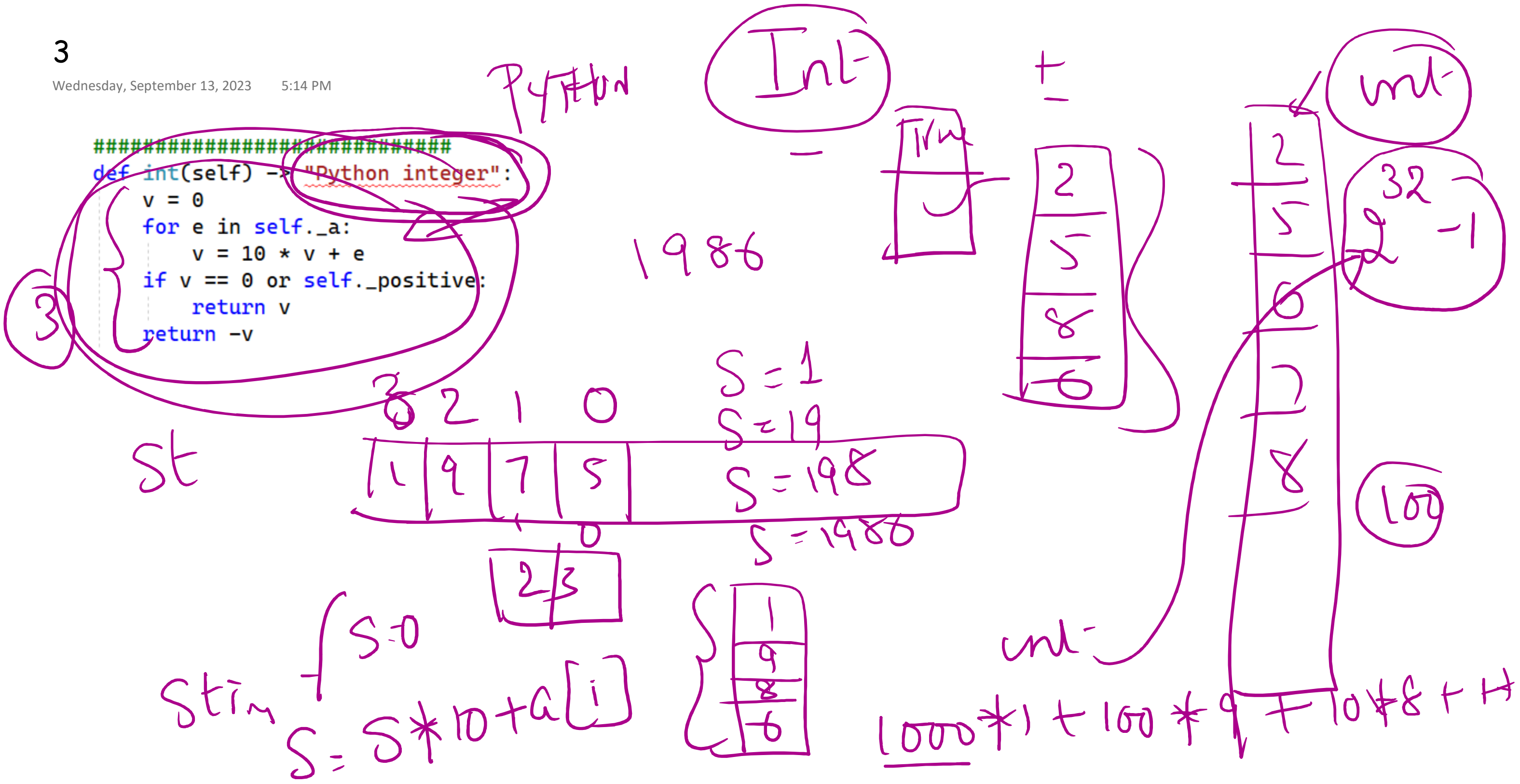
Wednesday, September 13, 2023 5:08 PM



```
###########################################################
#  class  Int
###########################################################
class Int:
    def __init__(self, n:"Python int" = 0):
        # ONLY DATA STRUCTURE ALLOWED
        # self._positive
        # self._a
        self._positive = True
        if n < 0:
            self._positive = False
        self._a = self.build(n)
```

3

Wednesday, September 13, 2023 5:14 PM



```
##############################
def int(self) -> "Python integer":
    v = 0
    for e in self._a:
        v = 10 * v + e
    if v == 0 or self._positive:
        return v
    return -v
```

0 1 2 3 4

a = lnt(2 5 ~~7~~ 8 9)

4

Operator overload

return a[2] int __getitem(self, 2)

pos val

a[2] = 4

int__getitem(int pos)

a len() __len__

getItem

a = Int(1986)

Print(a)

__str__

c = a+b

__add__

5

Wednesday, September 13, 2023 5:22 PM

--le--

$<$

$a < b$

$a > b \equiv (b < a)$

$a < b$

$a \leq b$

$a > b$

$a \geq b$

$a == b$

$a != b$

--le--

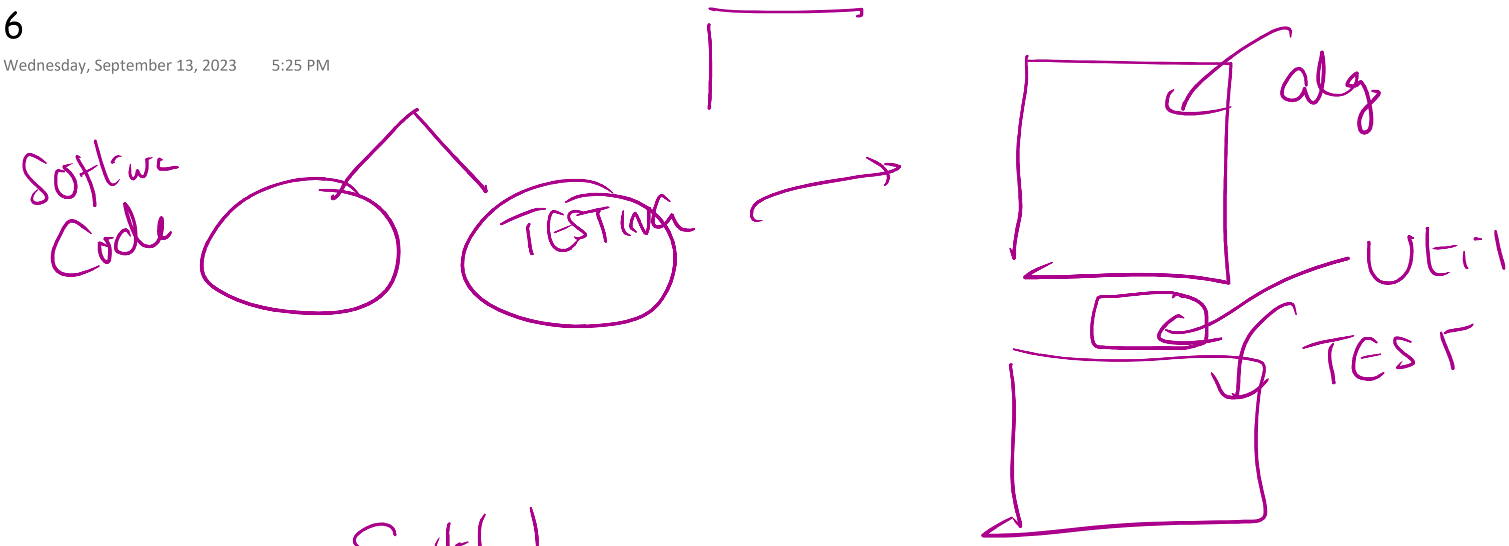
$a == b \quad !(a < b) \&\& !(b < a)$

Equal

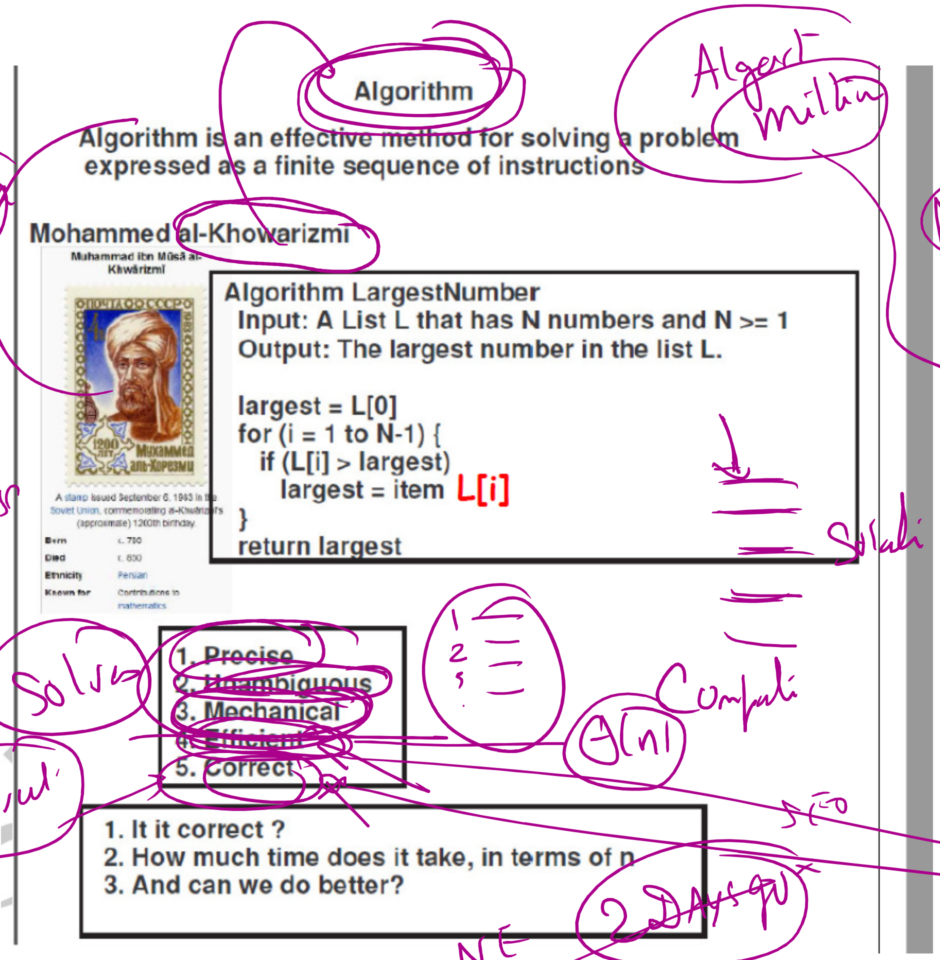
$a != b \quad !(a == b)$

6

Wednesday, September 13, 2023 5:25 PM



Sort( )



Algorithm

Algorithm is an effective method for solving a problem expressed as a finite sequence of instructions

Mohammed al-Khowarizmi

Muhammad ibn Mūsā al-Khwārizmī

A stamp issued September 6, 1983 in the Soviet Union, commemorating al-Khwārizmī's (approximate) 1200th birthday.

Born	c. 780
Died	c. 850
Ethnicity	Persian
Known for	Contributions to mathematics

```
Algorithm LargestNumber
  Input: A List L that has N numbers and N >= 1
  Output: The largest number in the list L.

  largest = L[0]
  for (i = 1 to N-1) {
    if (L[i] > largest)
      largest = item  L[i]
  }
  return largest
```

1. Precise
2. Unambiguous
3. Mechanical
4. Efficient
5. Correct

1. It it correct ?
2. How much time does it take, in terms of n
3. And can we do better?

100

Algeri milton

Computi

GPU

Nobel 1947

Bell labs

Transist

a — a

1 — ON

0 — 0ff

1/0

Persia

Given

Solati

Machine Lear

1971 Intel 4 bit

3 2 1 0

15

1973 Intel 8085

Solve

1 2 3

O(n)

Computi

Cort

sFo

GPS

efficient

Co

2 Days 90

NE

SFO Airport

SINE

Dijksta 1811

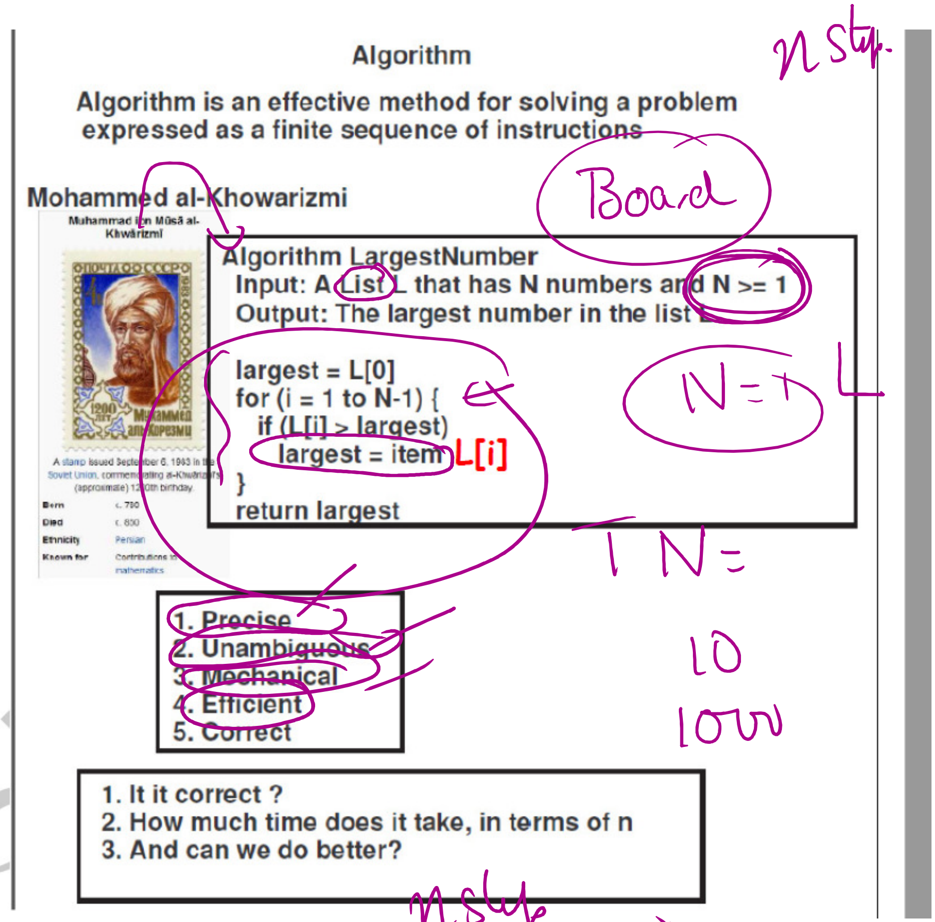
101 N

1. Make a left
2. Take 101 N
3. Drive 5.8
4. Exit 21 D

LOS

8

Wednesday, September 13, 2023 5:38 PM



Algorithm

Algorithm is an effective method for solving a problem expressed as a finite sequence of instructions

Mohammed al-Khowarizmi

Muhammad ibn Mūsā al-Khwārizmī

A stamp issued September 6, 1983 in the Soviet Union, commemorating al-Khwārizmī's (approximate) 1200th birthday.

Born	c. 780
Died	c. 850
Ethnicity	Persian
Known for	Contributions to mathematics

```
Algorithm LargestNumber
  Input: A List L that has N numbers and N >= 1
  Output: The largest number in the list L

  largest = L[0]
  for (i = 1 to N-1) {
    if (L[i] > largest)
      largest = item L[i]
  }
  return largest
```

1. Precise
2. Unambiguous
3. Mechanical
4. Efficient
5. Correct

1. It it correct ?
2. How much time does it take, in terms of n
3. And can we do better?

n step

Board

N = 1

L

1 N =

10

1000

log n

n

0 100 99

12 | 180 | 1 | 99 | 94

MAx : 99

10

0

2 | 5 | 1 | 95 | 23 | | |

1 student

2

95 Largest

Largest n

n step

194 | 99 | | | 99

n steps

Lucky n steps

$\Theta(n)$

Unlucky n steps

99 n-1

return a[n-1] $\Theta(1)$

return a[0] $\Theta(1)$

Sort Ascend or descend

2

2 | 2 | 2

$O(1)$

n^2

n log n

n

n

Sort

n log n

All method

n

9

Wednesday, September 13, 2023 5:46 PM

2.2.1 Finding a number in an unsorted array

Finding a suitcase in an unsorted airport baggage carousel

O(n) ≤ n tim

≤ n

n = 3W

Black

Work

Binding

Luck

1

1 step

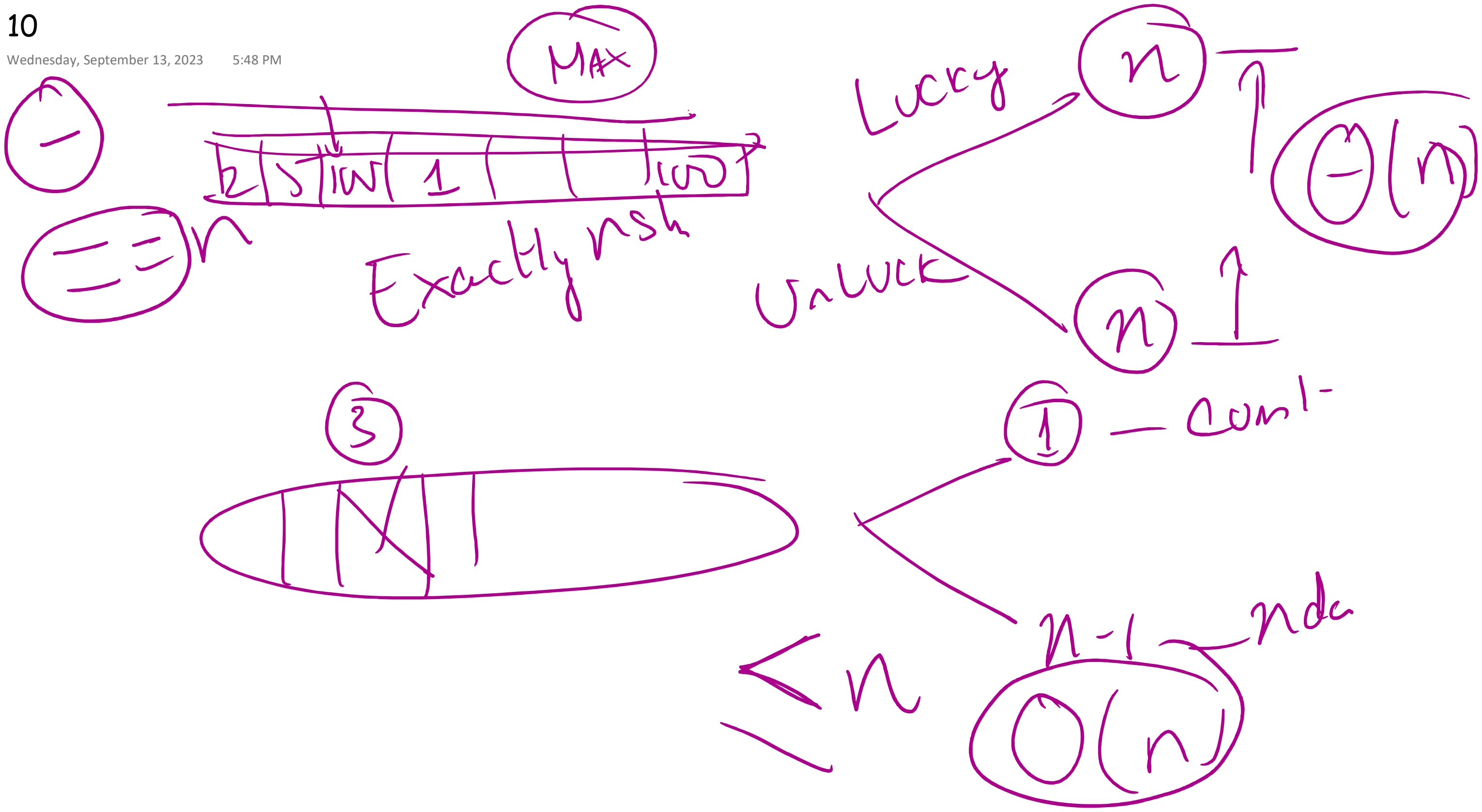
Unlucky

n

n step

10

Wednesday, September 13, 2023 5:48 PM



11

Wednesday, September 13, 2023 5:51 PM

n Coins

1$ — n = 1 mill

1 million step

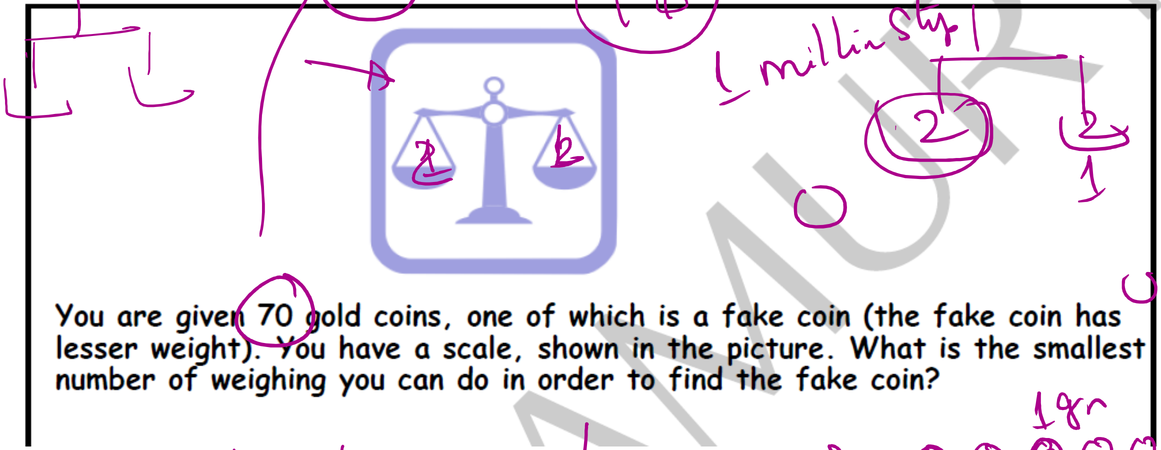
40 $

Luck 1

O(n)

O(1)

n



You are given 70 gold coins, one of which is a fake coin (the fake coin has lesser weight). You have a scale, shown in the picture. What is the smallest number of weighing you can do in order to find the fake coin?

2

2

1

FAKE 1

1 2gm

1gm

A

B

Lucky

1

2 | 2 | 2 | 2 | 2 | 2 | 2 | 1

Brute force

12

Wednesday, September 13, 2023 5:55 PM

Alg

n 16 → $n/2^0$

↓

8 $n/2^1$

↓

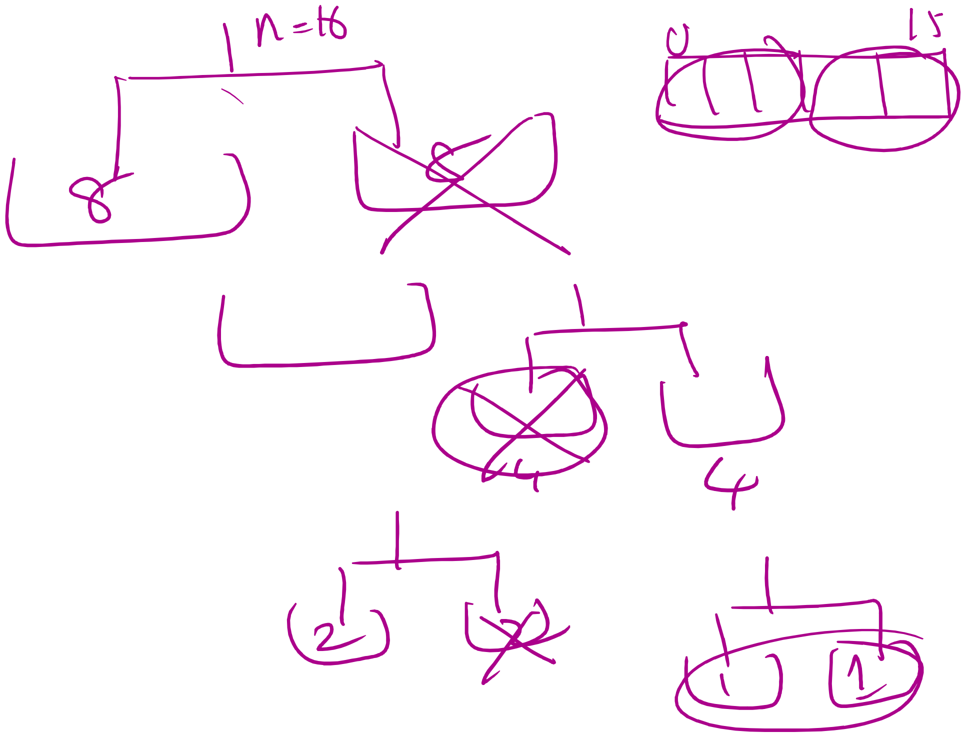
K 4 $n/2^2$

↓

2 $n/2^3$

↓ ↓

1 $n/2^K = 1$



Wednesday, September 13, 2023 5:58 PM

3

$\frac{16}{2}{}^{2}\ \frac{8}{2}{}^{4}\ \frac{4}{2}\ {}^{2}/ : 1$

$$\frac{n}{2^k} = \frac{1}{K}$$

$$K \left[\begin{array}{c} n/2^0 \\ \downarrow \\ n/2^1 \\ \downarrow \\ n/2^2 \\ \downarrow \\ \vdots \\ n/2^k = 1 \end{array}\right.$$

$$\boxed{\log_2 n}$$

$$n = 2^K$$

$$\log_2 n = K \boxed{\log_2 2}$$

$$\boxed{K = \log_2 n}$$

HOW MANY TIME I HAVE TO DIVIDE n by 2 to get 1

14

Wednesday, September 13, 2023 5:59 PM

n	n	Log n	
16	16	4	$(2^4 = 16)$
100	100	$\boxed{6.7}$	$(2^6 = 64$ $2^7 = 128)$
1000	1000	$\boxed{10}$	$2^{10} = 1025$
1 million	1 milli	$\boxed{20}$	$2^{20} \approx$ 1 milli
1 Bill	— 1 Bill	$\boxed{30}$	$2^{30} \approx$ 1 Billi

$\frac{\cancel{16}}{\cancel{2}}$ $\frac{\cancel{8}}{\cancel{2}}$ $\frac{\cancel{4}}{2}$ $\frac{2}{2}$

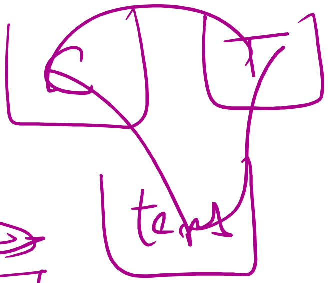
15 Polyno

Wednesday, September 13, 2023 6:01 PM

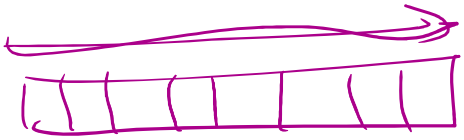
1. $\Theta(1)$ Alg
2. $\log n$
3. n
4. $n \log n$
5. n^2
6. n^3

n^4 n^6

1. 3 steps Constant Space



1 Billion $n = 30$ steps



$2^{10} = 1024$

$n \log n$

$1000 * 10$

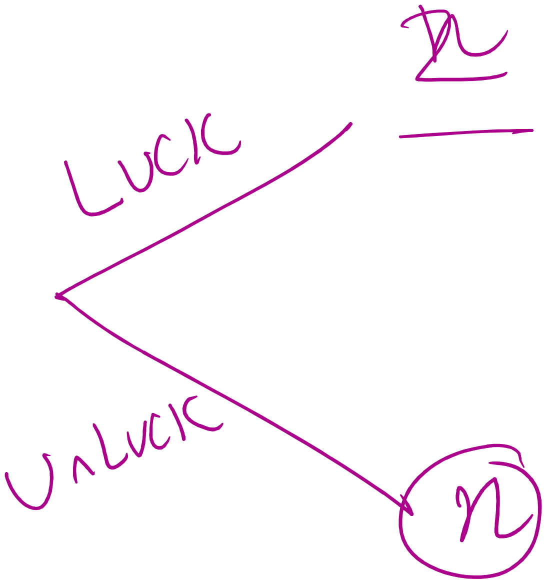
10,000

n^2

$1000 * 1000 = 1$ mill

n^2

$1, \log n, n, n\log n, n^2, n^3, \ldots n^6, 2^n, 3^n \ldots !n$



$n = 64$

$\Theta(n)$

$C \le n$ step $O(n)$

$< n$ step

17

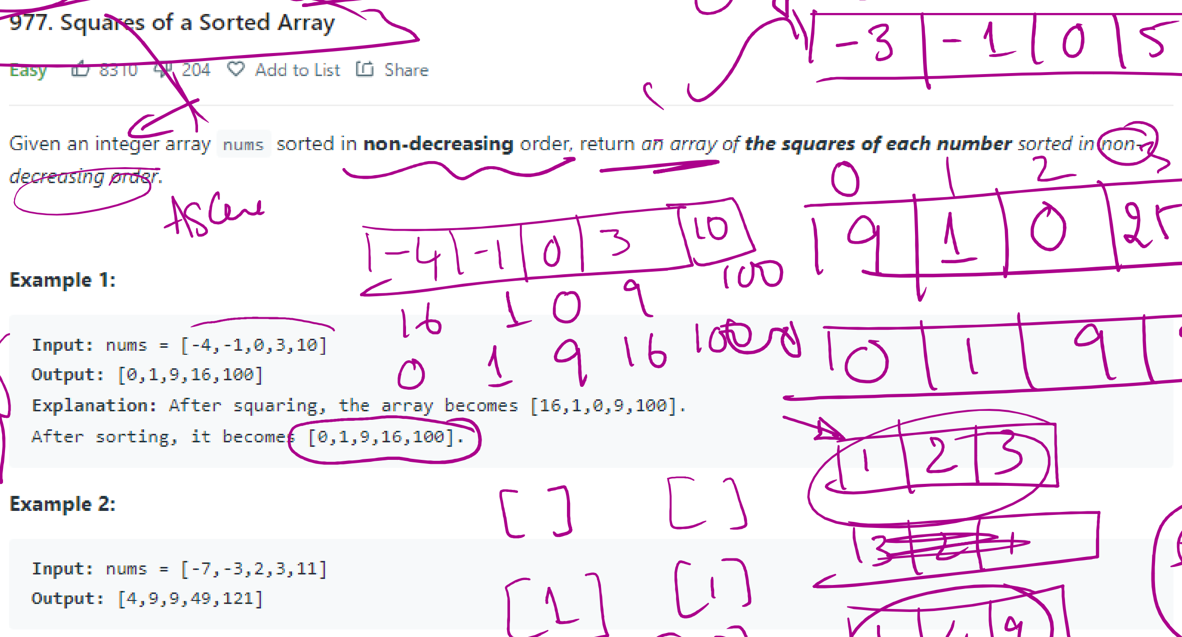
Wednesday, September 13, 2023 6:07 PM

LIST array

n-1

MS

n=5



977. Squares of a Sorted Array

Given an integer array `nums` sorted in **non-decreasing** order, return *an array of **the squares of each number** sorted in non-decreasing order*.

Ascen

Example 1:

```
Input: nums = [-4,-1,0,3,10]
Output: [0,1,9,16,100]
Explanation: After squaring, the array becomes [16,1,0,9,100].
After sorting, it becomes [0,1,9,16,100].
```

Example 2:

```
Input: nums = [-7,-3,2,3,11]
Output: [4,9,9,49,121]
```

0 1 2 3 4

| -3 | -1 | 0 | 5 | 9 |

| -9 | 5 |

| 81 | 25 |

0 1 2 3 4

| 9 | 1 | 0 | 25 | 81 |

| -4 | -1 | 0 | 3 | 10 |

16 1 0 9 100

0 1 9 16 100

| 0 | 1 | 9 | 25 | 81 |

| 1 | 2 | 3 |

| 1 | 4 | 9 |

[ ] [ ]

[1] [1]

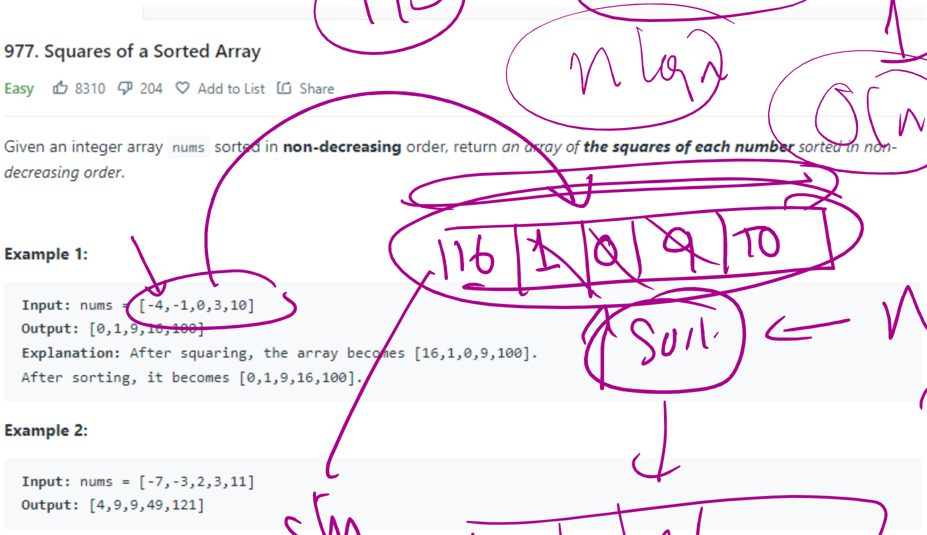
[10] [100]

-3 9

+3 9

18

Wednesday, September 13, 2023 6:14 PM



977. Squares of a Sorted Array

Easy 8310 204 Add to List Share

Given an integer array `nums` sorted in **non-decreasing** order, return *an array of **the squares of each number** sorted in non-decreasing order.*

Example 1:

```
Input: nums = [-4,-1,0,3,10]
Output: [0,1,9,16,100]
Explanation: After squaring, the array becomes [16,1,0,9,100].
After sorting, it becomes [0,1,9,16,100].
```

Example 2:

```
Input: nums = [-7,-3,2,3,11]
Output: [4,9,9,49,121]
```

MS

PYTHON Sort

n log n

1 milli

O(n)

| 16 | 1 | 0 | 9 | 100 |

Sort

nlogn

nlogn

O(n)

| 0 | 1 | 9 |

O n sp

nlogn sp

nlogn

Brute-force

S 1

a = 1986

a[2]

Strs = "1986"

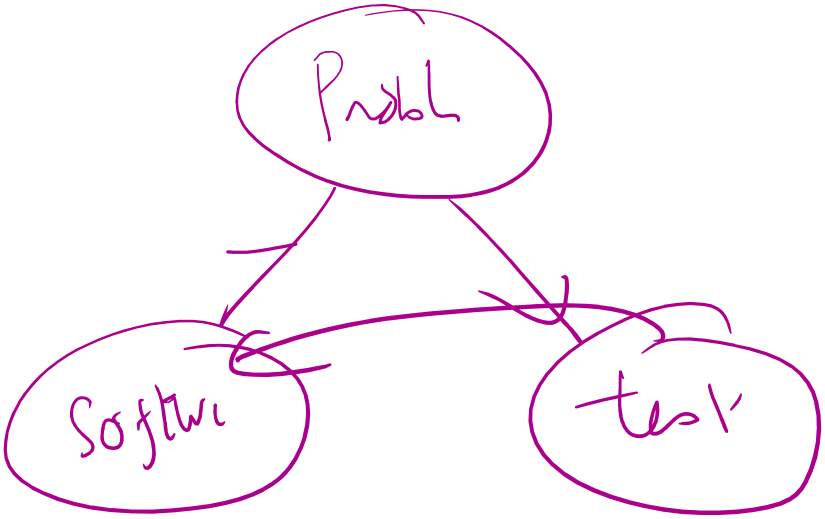
s[2]

10) 1986 (

0.9

19

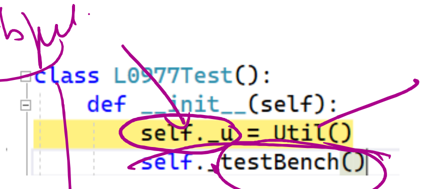
Wednesday, September 13, 2023 6:19 PM



20

Wednesday, September 13, 2023 6:24 PM

Objec.



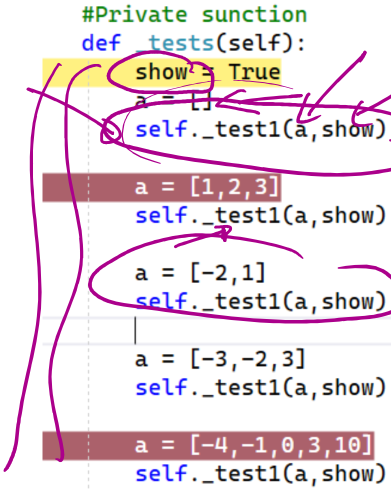
```
class L0977Test():
    def __init__(self):
        self._u = Util()
        self._testBench()
```

class

Self._U(   )

≡ Alg

0



```
#Private sunction
def _tests(self):
    show = True
    a = []
    self._test1(a,show)

    a = [1,2,3]
    self._test1(a,show)

    a = [-2,1]
    self._test1(a,show)

    a = [-3,-2,3]
    self._test1(a,show)

    a = [-4,-1,0,3,10]
    self._test1(a,show)
```

[ ] [ ]

[1, 4, 9]

-2, 1

4, 1

1, 4

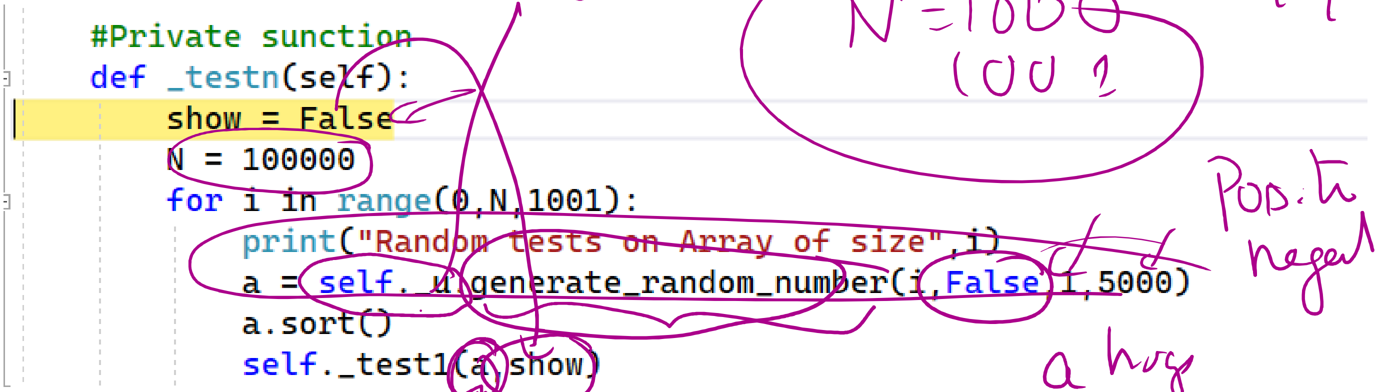
-3, -2, 3

9 4 9

4 9 9

21

Wednesday, September 13, 2023 6:29 PM



```
    #Private sunction
    def _testn(self):
        show = False
        N = 100000
        for i in range(0,N,1001):
            print("Random tests on Array of size",i)
            a = self._u_generate_random_number(i,False,1,5000)
            a.sort()
            self._test1(a,show)
```

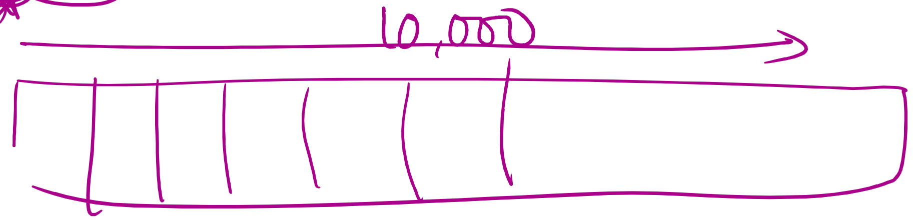
Object- Class

N = 1000
1001 ?

-919 -256

Positive negative

a huge

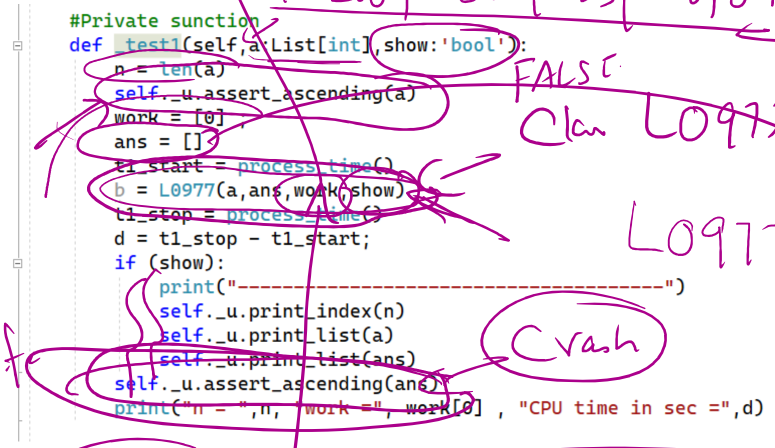


22

Wednesday, September 13, 2023 6:35 PM

Complexity

| -200 | -88 | -33 | -20 | 0 | 25 | 75



```
#Private sunction
def _test1(self,a:List[int],show:'bool'):
    n = len(a)
    self._u.assert_ascending(a)
    work = [0]
    ans = []
    t1_start = process_time()
    b = L0977(a,ans,work,show)
    t1_stop = process_time()
    d = t1_stop - t1_start;
    if (show):
        print("---------------------------------------")
        self._u.print_index(n)
        self._u.print_list(a)
        self._u.print_list(ans)
    self._u.assert_ascending(ans)
    print("n = ",n, "work =", work[0] , "CPU time in sec =",d)
```

FALSE

Clean L0977

Verify

L0977(a, ans, work, Show)

Crash

Verify

19

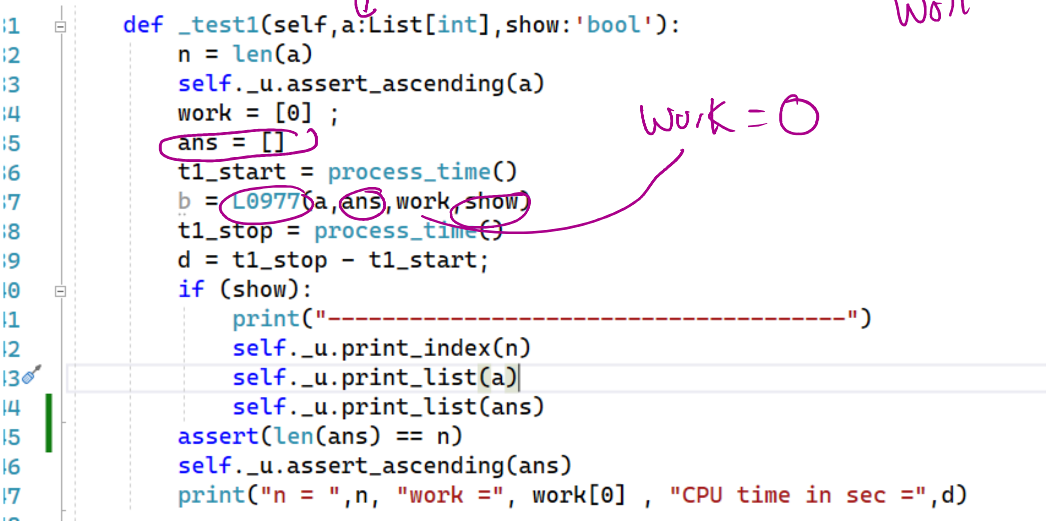
Complexity Verification

6:55 PM

Record

23

Wednesday, September 13, 2023 6:56 PM



```
def _test1(self,a:List[int],show:'bool'):
    n = len(a)
    self._u.assert_ascending(a)
    work = [0] ;
    ans = []
    t1_start = process_time()
    b = L0977(a,ans,work,show)
    t1_stop = process_time()
    d = t1_stop - t1_start;
    if (show):
        print("--------------------------------------")
        self._u.print_index(n)
        self._u.print_list(a)
        self._u.print_list(ans)
    assert(len(ans) == n)
    self._u.assert_ascending(ans)
    print("n = ",n, "work =", work[0] , "CPU time in sec =",d)
```

Work = 0

Woil

Work = 0

Work = S

W '

Wo

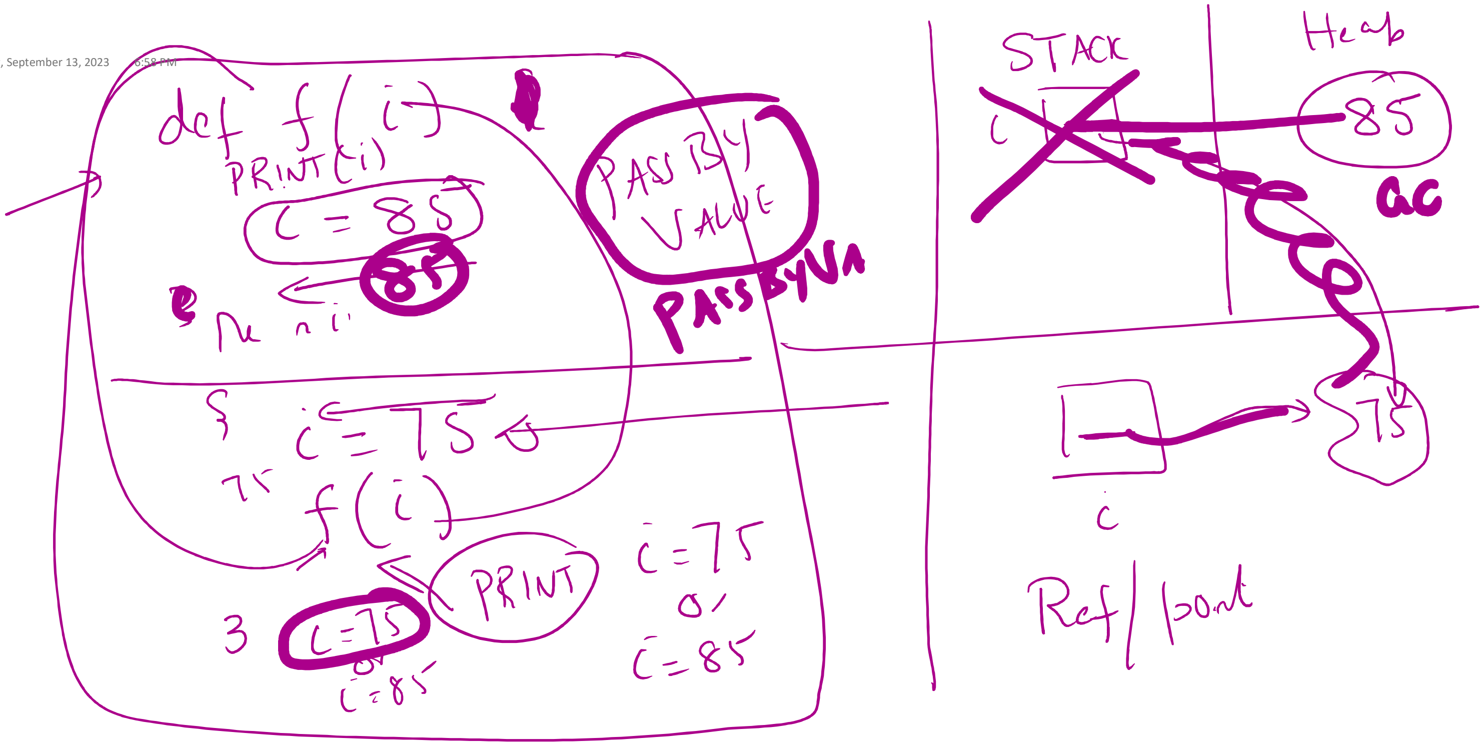
~~Work~~[

Work = [0]

work = 0

24

Wednesday, September 13, 2023 6:58 PM



25

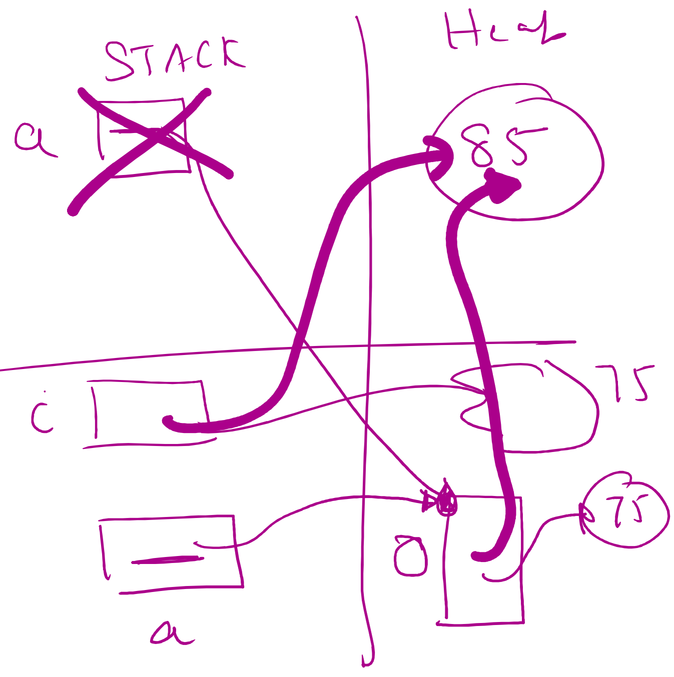
Wednesday, September 13, 2023 7:02 PM

PAR

```
f(a: 'List of Size 1')
    PRINT(a[0])
    a[0] = 85
    return
i = 75
a = [i]
f(a)
```

JAVA

i is 75
i = a[0]
i : 85



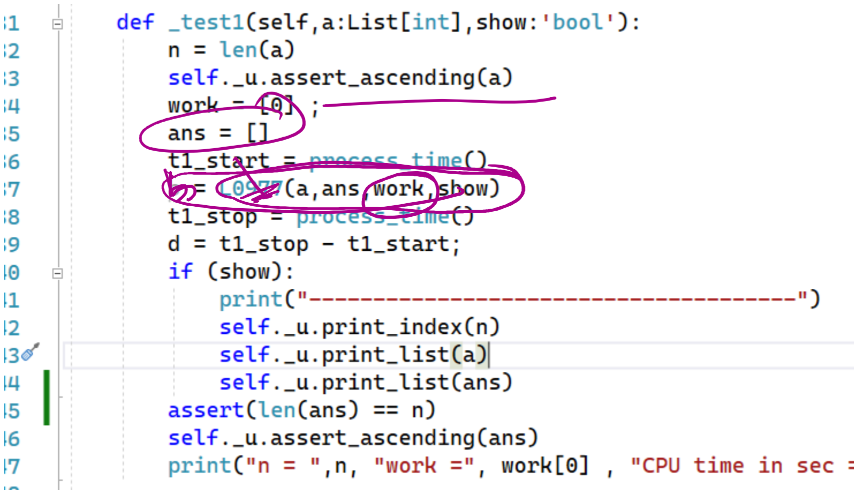
26

Wednesday, September 13, 2023 7:05 PM

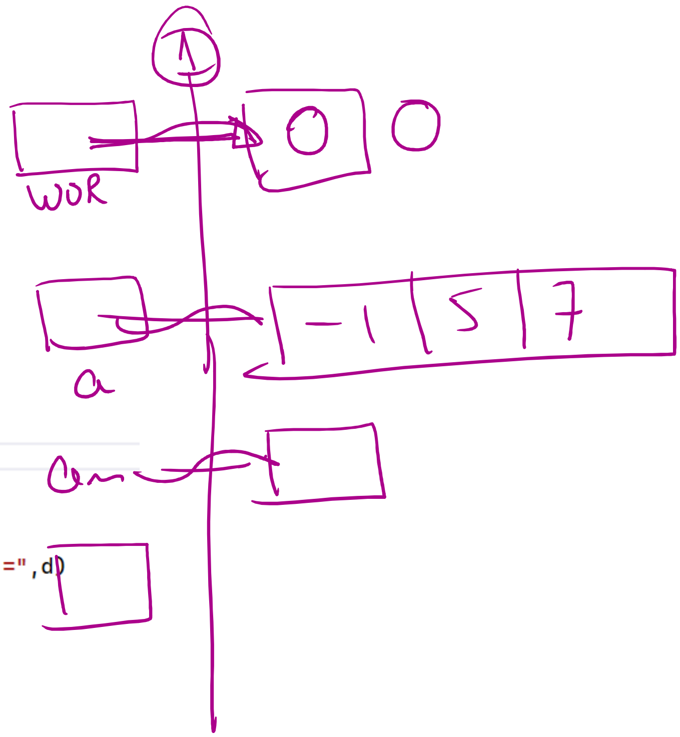
Constructors CAN'T Return Object

27

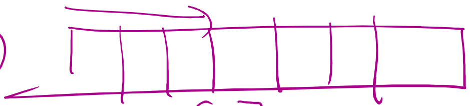
Wednesday, September 13, 2023 7:05 PM



```
def _test1(self,a:List[int],show:'bool'):
    n = len(a)
    self._u.assert_ascending(a)
    work = [0] ;
    ans = []
    t1_start = process_time()
    b = L0977(a,ans,work,show)
    t1_stop = process_time()
    d = t1_stop - t1_start;
    if (show):
        print("-------------------------------------")
        self._u.print_index(n)
        self._u.print_list(a)
        self._u.print_list(ans)
    assert(len(ans) == n)
    self._u.assert_ascending(ans)
    print("n = ",n, "work =", work[0] , "CPU time in sec =",d)
```



Single dual tree



n^2

a

n

[-1,2,100)

[ ]

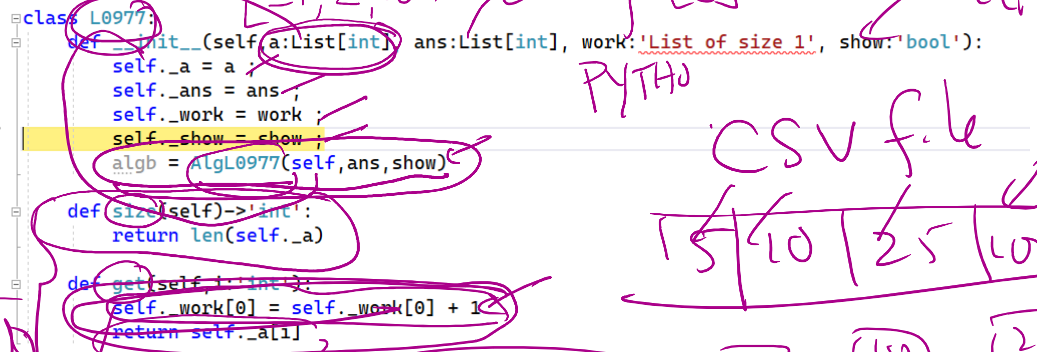
[0]

PYTHO

n

95

$n=5$



```
class L0977:
    def __init__(self,a:List[int], ans:List[int], work:'List of size 1', show:'bool'):
        self._a = a ;
        self._ans = ans ;
        self._work = work ;
        self._show = show ;
        algb = AlgL0977(self,ans,show)

    def size(self)->'int':
        return len(self._a)

    def get(self,i:'int'):
        self._work[0] = self._work[0] + 1
        return self._a[i]
```

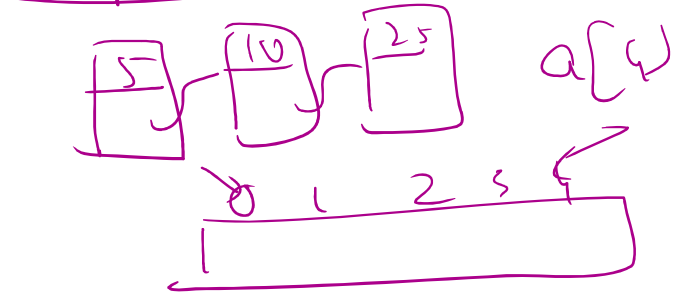
CSV file

5 | 10 | 25 | 100

get(

get(3)

work = work +1

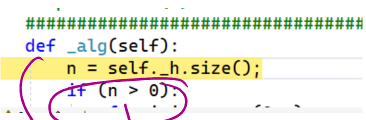


29

Wednesday, September 13, 2023 7:14 PM

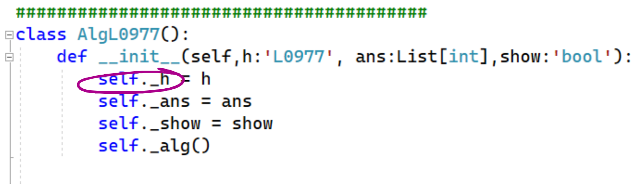
29

n line



```
#################################
def _alg(self):
    n = self._h.size();
    if (n > 0):
```

10 lines of code



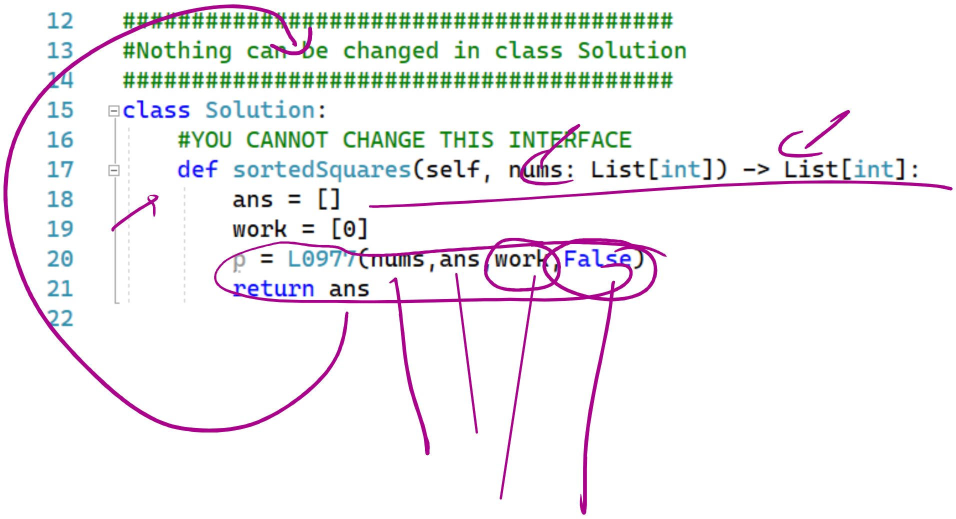
```
########################################
class AlgL0977():
    def __init__(self,h:'L0977', ans:List[int],show:'bool'):
        self._h = h
        self._ans = ans
        self._show = show
        self._alg()
```

n = self._h.size()

= self._h.get(3)

30

Wednesday, September 13, 2023 7:17 PM



```
########################################
#Nothing can be changed in class Solution
########################################
class Solution:
    #YOU CANNOT CHANGE THIS INTERFACE
    def sortedSquares(self, nums: List[int]) -> List[int]:
        ans = []
        work = [0]
        p = L0977(nums,ans,work,False)
        return ans
```

an = [ ]

work = (0)