~~Day~~ class-2

7/31/21

DSA

Big O
↓
worst case

~~ath~~

Reverse → (array) → $\overset{T}{O}(n)$

Space: O(1)

String → "mom", "LOL" (symmetrical)

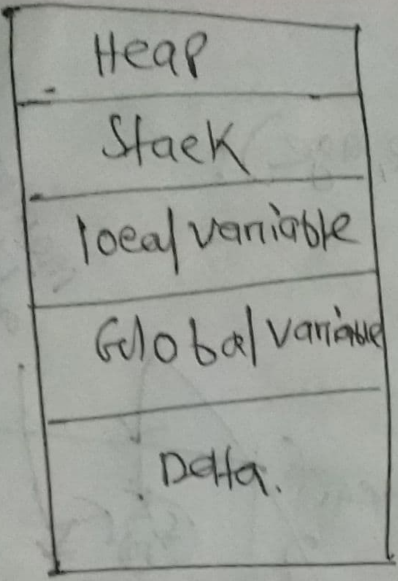
S_1 →	M	O	M
S_2 →	M	O	M

O(n)

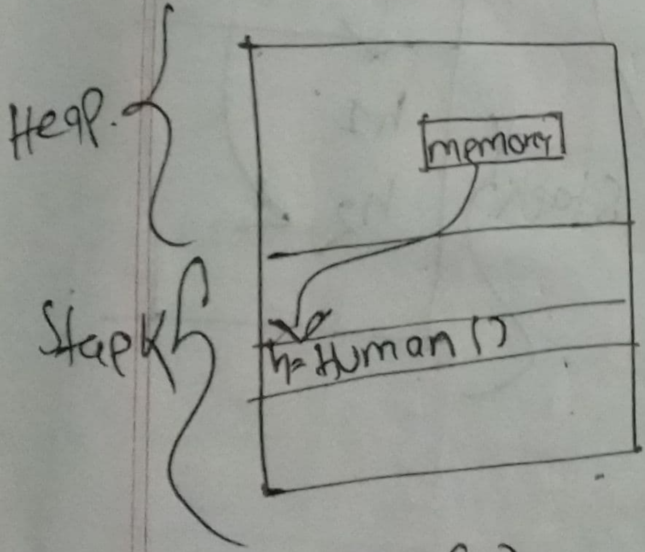
O(1) — constant

class: Blue print

class solution:



PCB



print(hi).

* সব Object তৈরি হয় heap এ.

* Object এর Refs তৈরি হয় stack এ.

* class design করতে, প্রয়োজন -

1. এখানে attribute = charaeteristies
2. method = behaviour
3. Construetion..

Object = Container

```
def __init__(self, )

class Human:
    def __init__(self, name, age):
        self.name = name
        self.age = age

h = Human("sajal", 50).
```

class defination

// যখন object তৈরি করব, তখন Constructor call হবে।

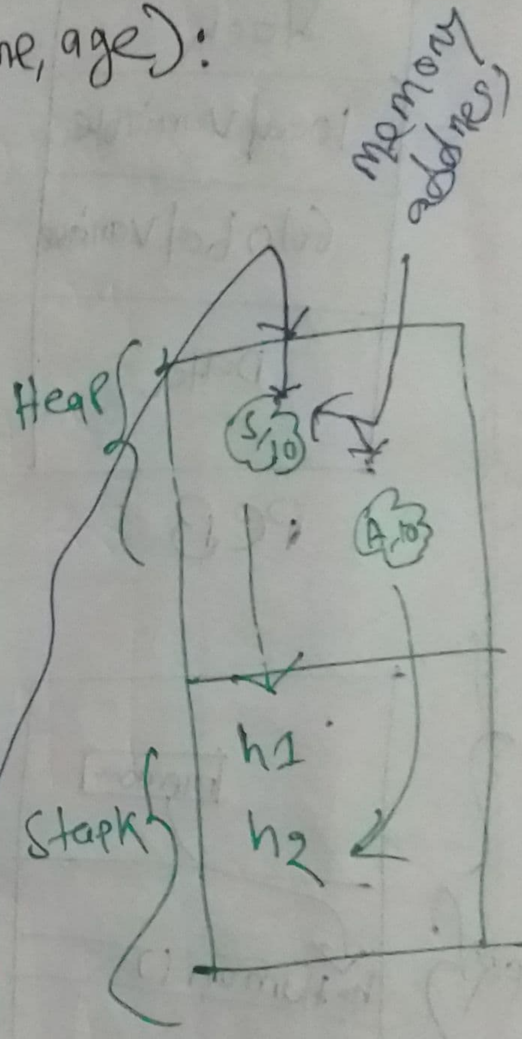
3

```
class Human:
    def __init__(self, name, age):
        self.name = name
        self.age = age.

    def getAge (self):
        return self.age.


h1 = Human ("Shazal", 50).
h2 = Human ("Ataur", 60)

                    1
a = h1.getAge ()
b = h2.getAge().
```



8

leet code

```
def □
class Solution:
    def □--init--□(self,□):
        self.□ =name
        self.□ = age.
```

leet code 125

```
class Solution:
    def isPalindroom(self, s: str)>bool:
        for char in s:
            //print(char) //

        start = 0
        end = len(s)-1
        While start<end:
```

m	A	D	A	m
0	1	2	3	4.

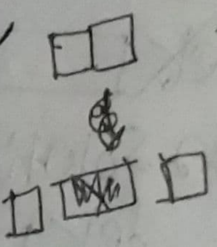
m	m
0	1

t	e	n	e	n	t
0	1	2	3	4	5.

5

m	A	D	A	M

→ length 5 →



s:	e:	a[s]	a[e]	s<e
0	4	M	M	0<4
1	3	A	A	0<3
2	2			2<2

```
Start = 0
end = len(S) - 1
While Start < end:
    If S[Start] != S[end]
        return False

    Start += 1
    end -= 1

return True.
```

start	end	start<end			
0	4	0<4	T	M	M
1	3	0<3	T	A	A
2	2	2<11			

A	M	M	A
0	1	2	3.

S<e   S see

167 leetcode two Sum II

Give anr Sortted arrray.

* Find target that can Sum up two numbers.

[2, 7, 11, 15] Sorted List

target=9

Hashmap : T O(n)

- S O(n).

[2, 7, 11, 15]   T=9

~~T=17~~ 17

→ Greater

← less

I = 0 ....len(a) -1

J = len(a) -1 0

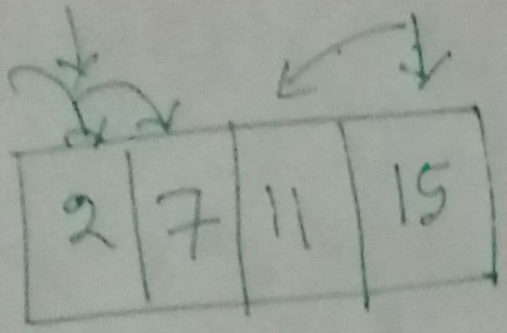
a_0 — a_1

a_0........ a_1.

$a_I + a_J$

$I < J$

$l = 0 ..... 1.$



2	7	11	15

Target

18

7+11

18

$a_i + a_j < t$

i নিয়ে কাজ করব

$a_i + a_j > t$

j নিয়ে কাজ করব

* [2,7,11,15]

j-2,

7 in a → (o n).

if t9-num1 in a: o(n)

return [ ] o(n^2).

why hashmap o(1)