D-31-08 class - 2

Space C → O(1)

T C → O(1) is possible ?

depends on • DSA

search → (Data structure)

item in lists → O(n)

item in dict → O(1)

item in my

efficient ← prev sol'3

reverse → array T : O(n) | O(n)

S : O(1) | O(n)

(Two pointer approach)

Palindrom: MOM , LOL , MADAM

s → ← e

.. | m | o | m |

match while moving

s1 | m | o | m |

s2 | m | o | m |

S → O(n)

python - oop

class - obj oo blueprint

class Human {

Name : S

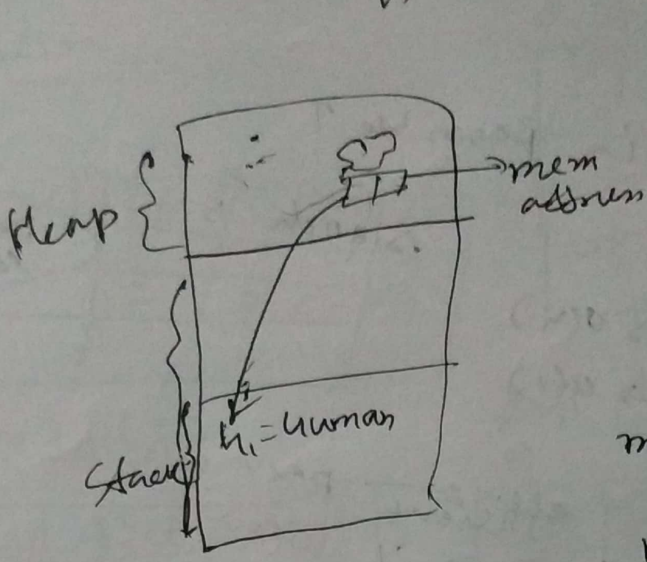
Age : I

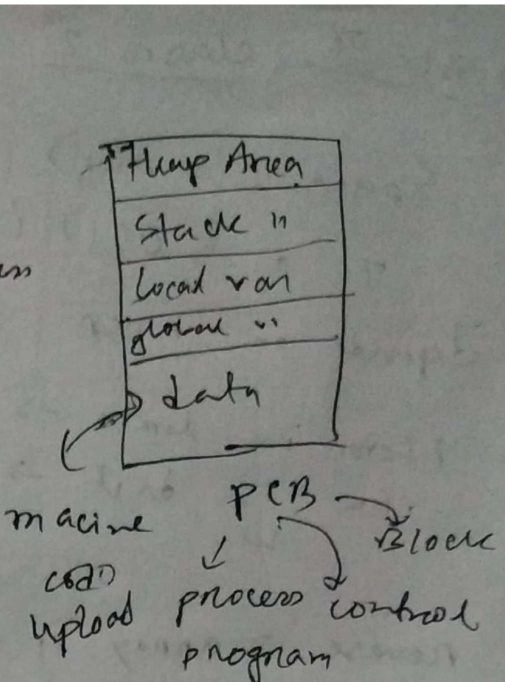
Gender : S

}

h1 = Human()

75>60





heap = Object create

stack = " ref.

Atribute – class property/characteristic/noun

method – behaviour/action/verb

constructor –

class

def __init__(self, ...)
(dunder)

custom container
[list, dict, set)

mutable – state/content changed

immutable – ✗

{(string, int, float, bool, tuple)
primitive

class def

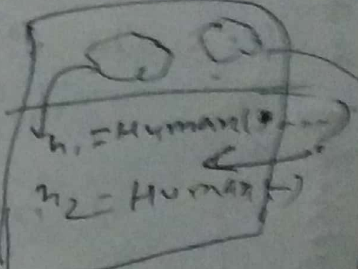
class current obj

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

Constr

self -> classname (obj)

```
h1 = Human("Nahid", 23)
```

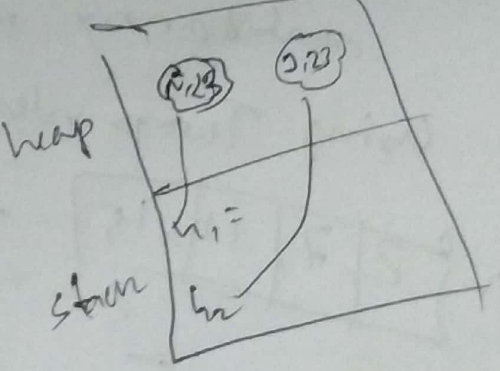


* why create obj?
=> obj in a container. store data

=> why constructor?
to store data . when initializing obj.

def walk() vs def walk(self)
class method

```
class Human:
    def __init__(self, name, age)
        self.name = name
        self.age = age
    def getAge(self):
        ~~return~~ self.age
        return
h1 = Human("Nahid", 23)
h2 = Human("Jakib", 23)
h1.getAge() . /
```



125. valid Palindrome

string → modify → ~~to~~ up → low
space → pop

M	A	D	A	M

a[s]

s	e	~~a[s]~~	a[e]	Matching	
s	e	a[s]	a[e]	s<e	M
0	4	M	M	0<4	L
1	3	A	A	1<3	W
2	2	/		2<2	~~[illegible]~~

string matching

```
start = 0
end = len(s) - 1
while start < end:
    if s[start] != s[end]
        return False
    start += 1
    end -= 1
return True
```

alphanumeric ~~[illegible]~~ char

'' . join(char for char in s if

167 - Two num II -
sorted array so,
- using Two pointer approach without dictio

2	7	11	15

a

t = 9

e ←

s →

a[s] + a[e] > t
e--
else s++

2 + 15 = 17 > 9
←

2 + 6 = 8 < 9
→

for i in a:
    if ·

target num

```
for i in a : O(n)  → ~~[illegible]~~
    if t-i in a : O(n)
        return [ ]
```
a → list

O(n²)

D-13/08

26 - remove duplicate.

T(1) - O(n)

S - O(1) ; same list এ করতে হবে,

S.case - len(list) → 0 হলে return 0

27 - remove element

same as 26

35 - search insert.

linear search → O(n)

Binary " → O(log n)

1	3	5	6
0	1	2	3

l r

~~mid = (left + righ) / 2~~

mid = ~~[illegible]~~ $\frac{right - left}{2}$ + left

66. Plus -one

[1,2,3] [4,3,2,1] [0] [1,9,9 ?]

↓ [1] [2,0,0 0

[1,2,4) [4,3,2,2]

19 (9+1) → 10 % 10

last = (last +1) % ? d[end] ↓ 0

9 9 e

8 9 9 9 e

↓ ↓ ↓

9 0 0 0

88 - merge sorted array / [1] [0]

[.1,2,3, 0,0,0] [2,56] ↓ (n) nums2 ↓ [1]

↓ nums1 ↓ [1,2,2,3,5,6] [0] [1]

m ↓ [1]

↓

O(n+m)

0≤m, n ≤ 200.

store in nums1.

problem - list index out of bounds handle