

DAA PY 5

Wednesday, October 4, 2023 4:43 PM

5 PM

JV

Jagadeesh Vasudevamu... (H

SC

Shrushti Chahande (Guest)

JC

Jingyi Chen

QC

Qian Chen

SC

Shrushti Chahande

ZC

Zhe Cao

JV

Jagadeesh Vasudevamurthy (Host, me)

AN

Anirudh Negi

CC

Claire (Yitong) Wu (Guest)

JZ

Jiading Zhou

JC

Jingyi Chen

ML

Mu Lyu

QC

Qian Chen

QY


Qing Yu

SC

Shrushti Chahande

SC

Shrushti Chahande (Guest)



Venni (cn: Wen Yu)

YJ

Yang Jiang

YL

Yuexin Li (Guest)

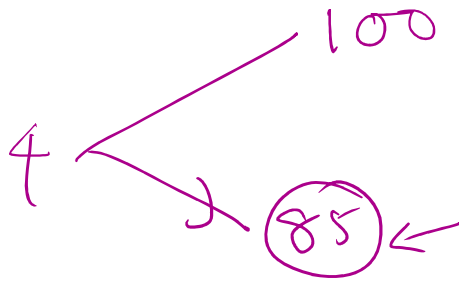
ZC

Zhe Cao

HS

Hongji Shi

7:25



5...

~~OCT-25 (Wed)~~
~~11P~~
~~13-4~~

WED

Zoom

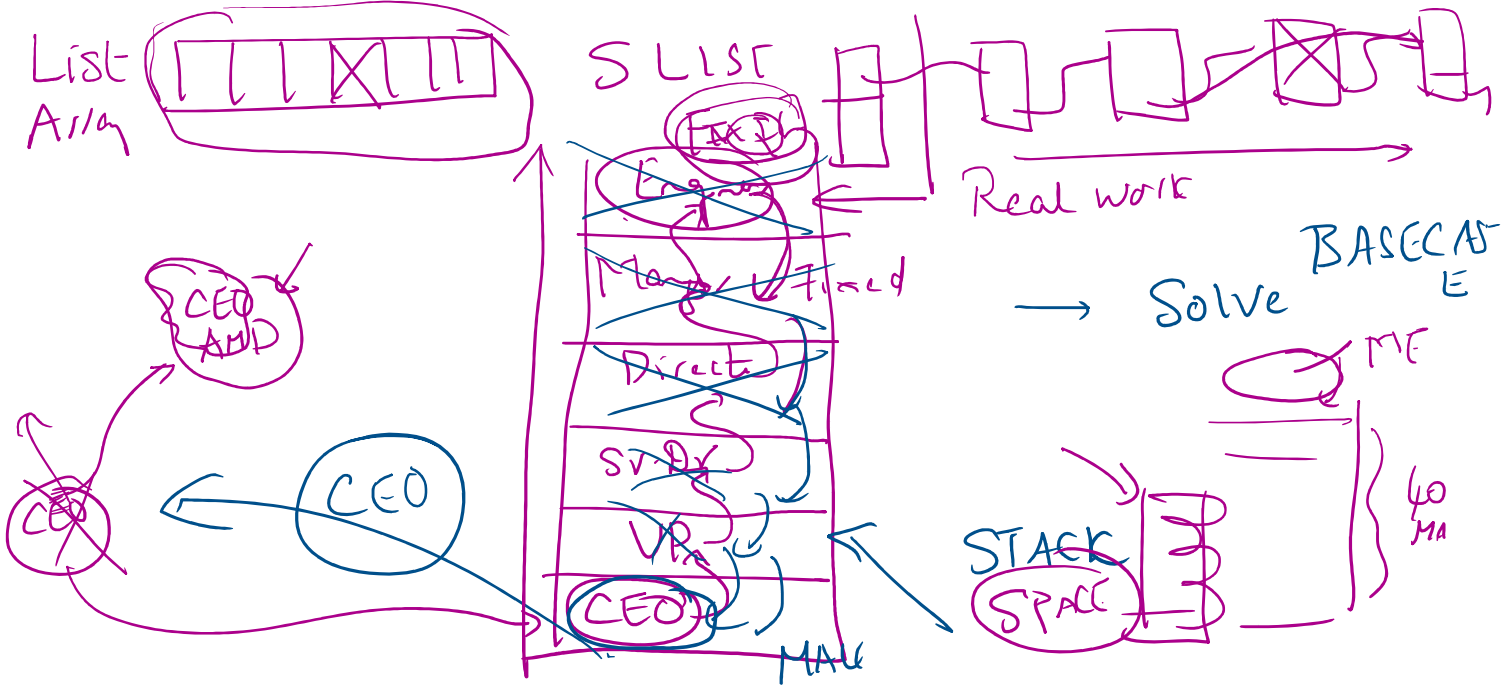
1 Student

9AM

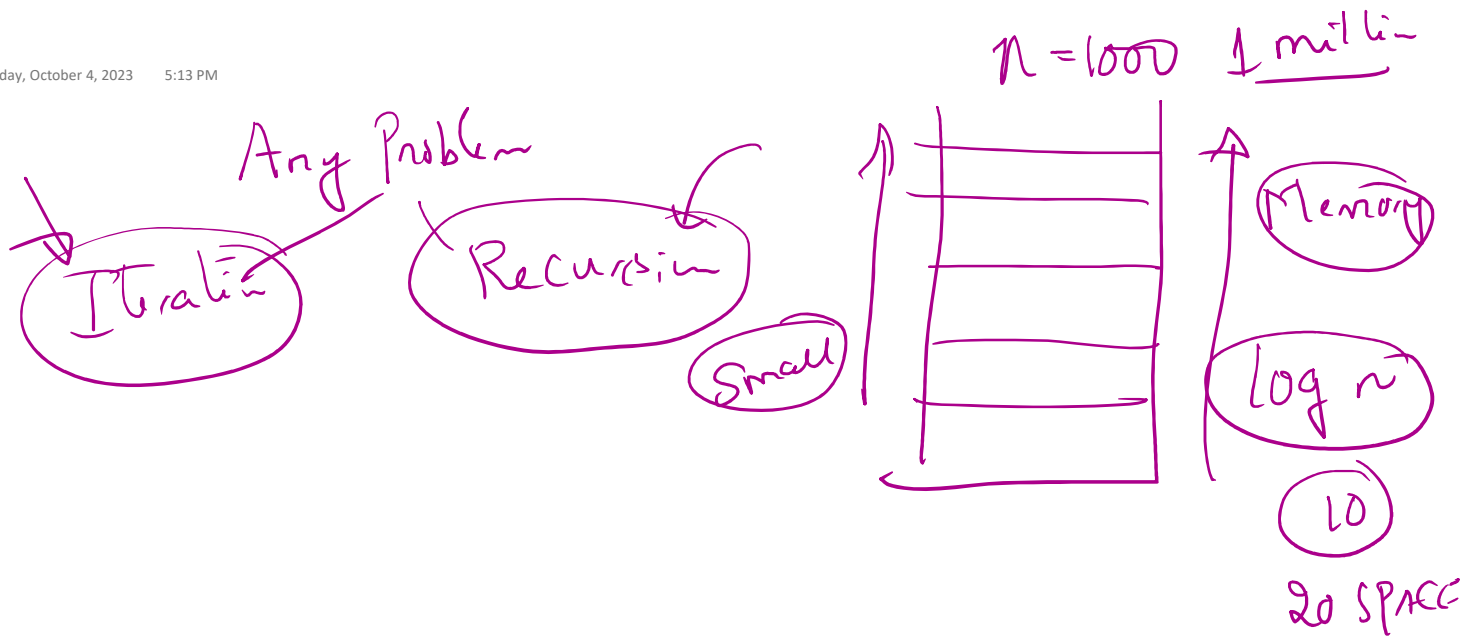
OCT-29

2 PM

Recursion



Wednesday, October 4, 2023 5:13 PM



Iteration

Recursion

Small

$n = 1000$ 1 million

Memory

$\log n$

10

2D SPACE

$n = 1000$
 $\text{int } F(\text{int } n) \{$
 $\quad \text{int } s = 1$
 $\quad \text{for } (i = 2; i \leq n; ++i) \{$
 $\quad \quad \text{Work} = \text{work} + 1$
 $\quad \quad s = s * i$
 $\quad \}$
 $\quad \text{return } (s)$
 $\}$

3
 3
 3

LUCKY
 UNLUCKY

$n = 4$
 $s = 1$
 $i = 2$
 $s = 2$
 $i = 3$
 $s = 6$
 $i = 4$
 $s = 24$

Time: $\Theta(n)$
 Space: $\Theta(1)$

$\{ \text{int } n = 5$
 $\text{int } a = \text{Fact}(n)$
 3

$n = 0$
 $n = 1$
 1
 1

$5 * 4 * 3 * 2 * 1$
 $24 * 5$
 120

int- $F(\text{int- } n)$

if ($n < 2$)
return 1

return ($n * F(n-1)$):

{ int- $n = 5$;
int- $a = F(n)$

→

$a = 120$

$1 = 1 = 1$
 $1 = 0 = 0$

1		
2	2	1
3	3	2
4	4	3
5	5	4

MULTIPLICATION

$n = 1000$ $n = 5$
 int F (int n) {
 if (n < 2)
 return 1
 return (n * F (n - 1));
 }

Time Complexity

3

{ int n	0	1	5
int a = F(n)	1	1	120
return a			

3

120

n=1			
n=2	2 * F(1)		
n=3	3 * F(2)		
n=4	4 * F(3)		
n=5	5 * F(4)		

n=5

1

997

998

999

1000

$$T(n) = T(n-1) + C$$

$S * 4$

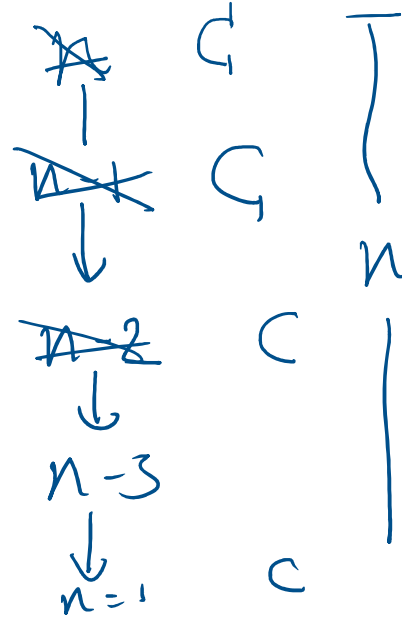
$$T(1) = 1$$

$$T(0) = 0$$

$$\text{Time} = Cn$$

$$= \Theta(n)$$

$$\text{SPACE} = \Theta(n)$$



void P(int n) {
 if (n == 0) PRINT(0); return;
 while (n) {
 PRINT(n % 10);
 n = n / 10;
 }
 }

Time: $O(\log_{10} n)$
 $O(1)$

1986
 1985
 1984

5

{ int n = 1986
 P(n)

6
 8
 9
 1

a = 11

3

5

n | ~~1986~~

6 8 9 1

n $\log_{10} n$
 ↓
 n/10
 ↓
 n/100
 ↓
 n/1000
 ↓
 1

1.

```
void R(int n) {
    if (n < 10)
        PRINT(n)
```

```
    else
        R(n / 10)
        PRINT(n / 10)
}
```

3

```
{ int n
  R(n)
```

3

```
0 1
0 1
```

1986

1

9

1

n = 1

n = 19

n = 198

n = 1986

198

R(1)	
P(19 - 1 * 10)	
R(19)	
P(198 - 1 * 10)	
R(198)	
PRINT(1986 / 10)	

1

9

8

6

```

void R(m, n) {
    if (n < 10)
        PRINT(n)
    else
        R(n / 10)
    PRINT(n / 10)
}

```

1 9 8 6

$n = 1986$

↓ ↓
 $n = 1986$
 1 9 8 6

SPACE

$n = 1$

$n = 19$

$n = 198$

$n = 1986$

10) 1986
 9 9
 8
 6

6 8 9

tober 4, 2023 5:34 PM

Void R (int- n) {
 if (n == 0) PRINT (0) return
 while (n) {
 PRINT (n % 10)
 n = n // 10
 }
 }

1986
 198
 19

3

1986
 198 19 1

n 1986

1 6
 9 8
 8 9
 6 1

Void R (int- n) {
 a = [] ← SPACE
 while (n) {
 a.append (n % 10)
 n = n // 10
 }
 }

3 3

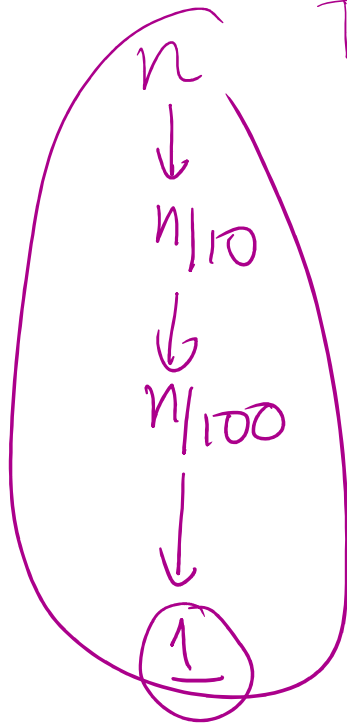
n 1986

a

0	1	2	3
6	8	9	1

10) 1986 / 198
 6

1
 9
 8
 6



$$T(n) = T\left(\frac{n}{10}\right) + C$$

$$T(0..9) = 1$$

$$T(n) = T(n-1)$$

$$\log_{10} n$$

SPACE

$$\log n$$

SPACE OF The stack

$\log n$

20

30

$R(m - n) \{ \begin{matrix} 0973876 \\ 10)1986 \end{matrix}$

$10)1986 (198 \quad 10)198 (19 \quad 10)19 (1$

$\begin{matrix} 10)10 \\ 9 \quad 1 \end{matrix}$

$\begin{matrix} 8 \\ 9 \end{matrix}$

$\begin{matrix} 6 \\ 8 \end{matrix}$

$\begin{matrix} 3 \\ 2 \\ 1 \\ 0 \end{matrix}$

$\begin{matrix} 6 & 8 & 9 & 1 \\ 1000 & 100 & 10 & 1 \end{matrix}$

$(6 * 1000) + (8 * 100) + (9 * 10) + (1 * 1)$

$24 \quad 3$

$10 \quad \{ \text{ml- } n = 1986$

$\text{ml- } a = R(n)$

$3 \quad \begin{matrix} 0 & 1 & 9 & 10 & 1000 & 1986 \\ 1 & 1 & 1 & 1 & 1 & \\ 0 & 1 & 9 & 1 & 1 & 6891 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 8 & 7 & 6 & & & & & \end{matrix}$

$\begin{matrix} 6 & 8 & 9 & 1 \end{matrix}$

$\text{int-} R(\text{int-} n)$
 $10) 1911$
 $\underline{9}$
 $10) 198(19)$
 $\underline{8}$
 $10) 19861$
 $\underline{6}$
 3
 $10) 1(0)$
 $\underline{1}$

 $\{ \text{int-} n = 1000 \quad 1986$
 $a = R(n)$
 $3 \quad (1) \quad 6891$

$a) \{ \begin{matrix} 1 & 2 & 3 \\ 0 & 1 & 2 \end{matrix} \}$
 $\downarrow \quad \downarrow \quad \downarrow$
 $6 \quad 8 \quad 9 \quad 1$
 $S = 0$
 $S = S * 10 + a[i]$
 $S \quad \boxed{6891}$
 $\boxed{}$

19867910
 int R (int ~~n~~)
 int S = 0
 while (n != 0) {
 $S = S * 10 + (n \% 10)$
 $n = n // 10$ Time: $\log_{10} N$
 }
 return S;
 SPACE: 2

{ $N = 1986$
 int a = R(N)
 }
 3

$N = 0$ $N = 1$
 $a = 0$
 6891 1

$10) 10000 (100$
 $\underline{0}$

n | 10000
 S | 01


```

int R(int n, int s) {
    s = (s * 10) + (n % 10);
    if (n < 10)
        return s;
    return R(n // 10, s);
}

```

$n = 1$
 $s = 689$

$n = 19$
 $s = 68$

$n = 198$
 $s = 6$

$s = 6891$

$s = 689$

$a = R(1, 689)$

return a

$s = 68$

$a = R(19, 68)$

return a

$s = 0 + 6 = 6$

$a = R(198, 6)$

return a

Help: int H(int n) {
 int s = 0;
 a = R(n, s);
 return a;
}

s 10

n 1989

Time: $O(\log n)$ $n = 1986$
 1986 Space: $O(\log n)$ $s = 0$

$n = 1986$

$a = R(n)$

3 **6891**

6891

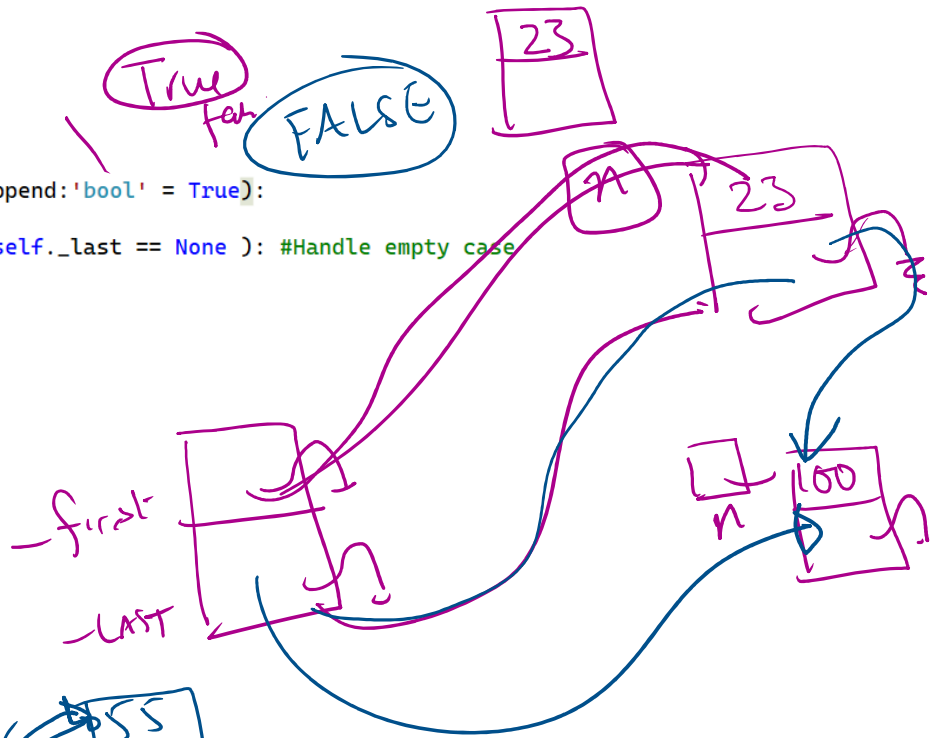
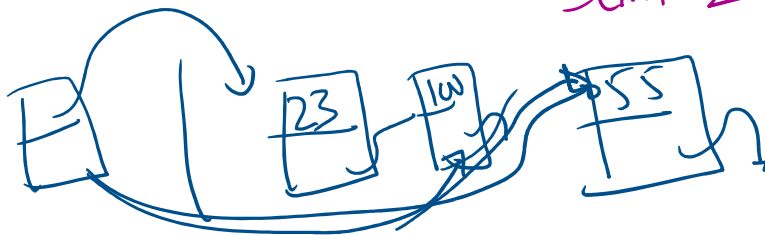
Object CLASS

```

def build_a_node(self, i: 'int', append: 'bool' = True):
    n = ListNode(i)
    if (self._first == None and self._last == None): #Handle empty case
        self._first = n
        self._last = n
    else:
        if (append):
            self._last.next = n
            self._last = n
        else:
            n.next = self._first
            self._first = n

```

True
False

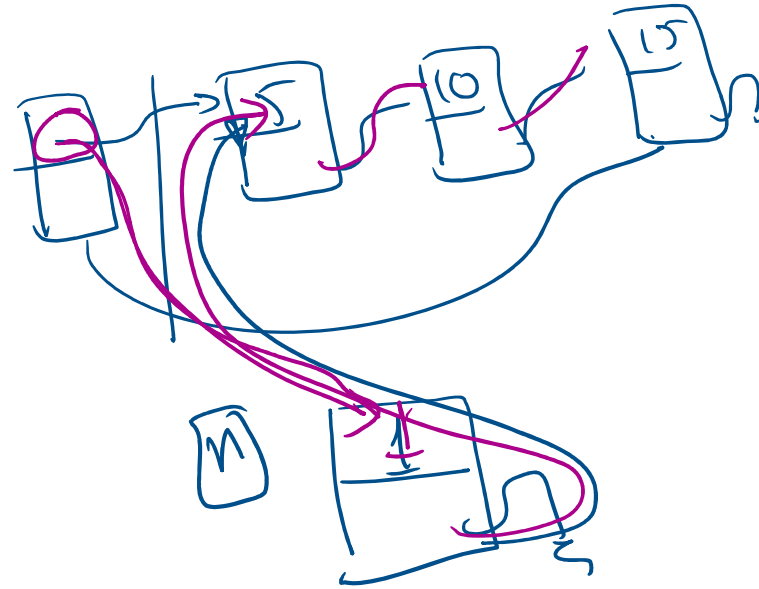


FALSE

```

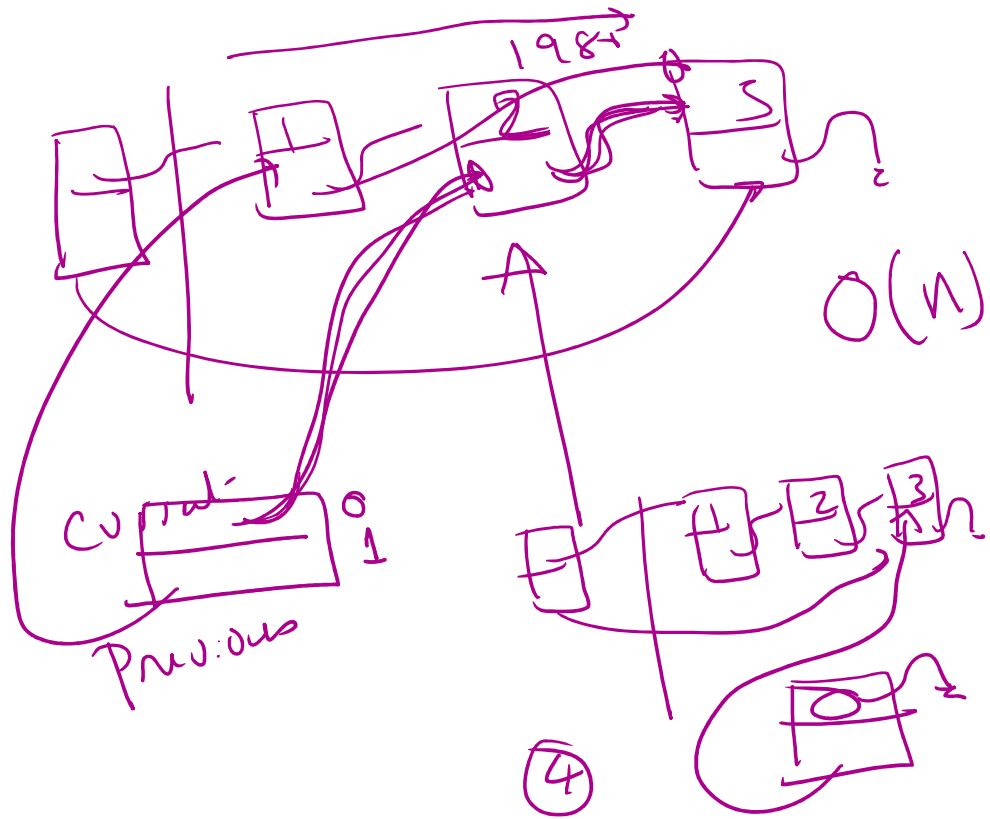
def build_a_node(self, i: 'int', append: 'bool' = True):
    n = ListNode(i)
    if (self._first == None and self._last == None ): #Handle empty case
        self._first = n
        self._last = n
    else:
        if (append):
            self._last.next = n
            self._last = n
        else:
            n.next = self._first
            self._first = n

```

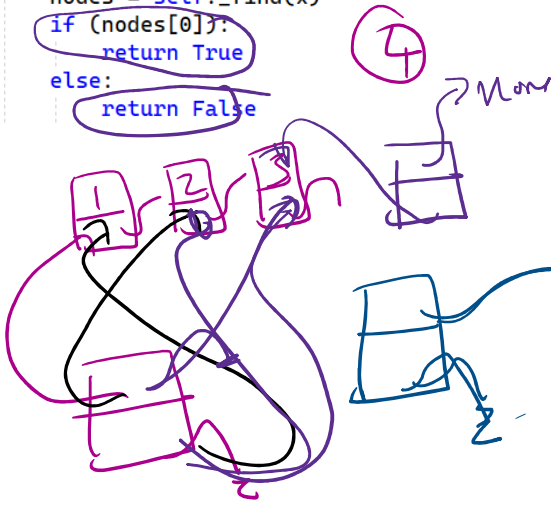
 $O(1)$ $O(1)$
SPACE

②

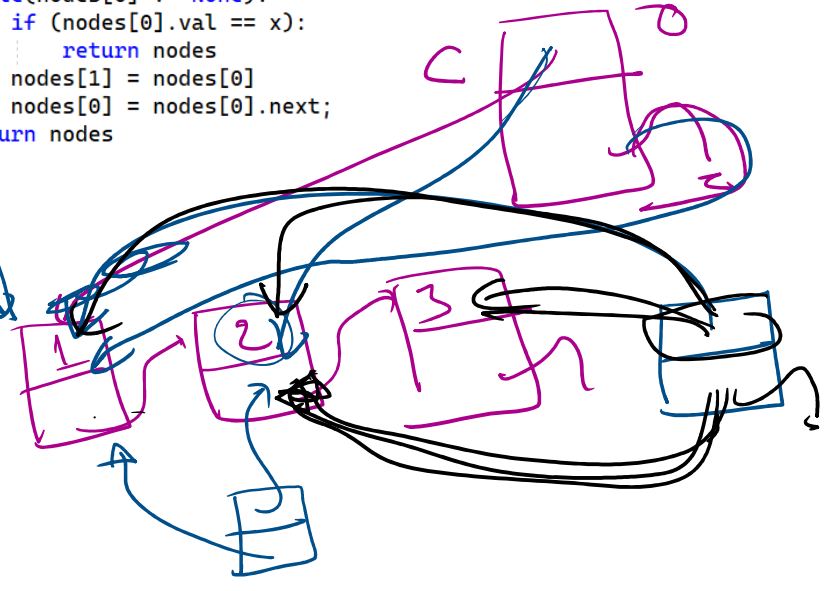
Delete



```
def find(self,x:'int'):
    nodes = self._find(x)
    if (nodes[0]):
        return True
    else:
        return False
```

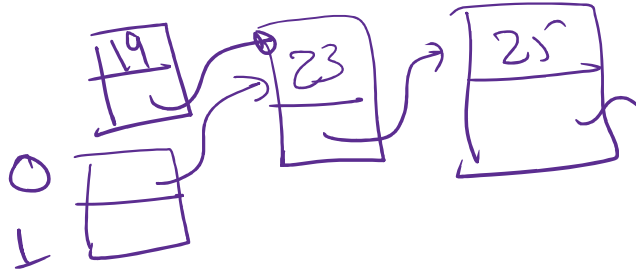


```
def _find(self,x:'int')->'list of [currentnode,prevnode]':
    nodes = [self._first,None]
    while(nodes[0] != None):
        if (nodes[0].val == x):
            return nodes
        nodes[1] = nodes[0]
        nodes[0] = nodes[0].next;
    return nodes
```



23

```
def delete(self,x:'int')->'bool':
    nodes = self._find(x)
    a = self._unhook(nodes)
    return a ;
```



```
def _unhook(self,nodes:'list of size 2')->'bool':
    if (nodes[0]):
```

```
        currentnode = nodes[0]
```

```
        previousnode = nodes[1]
```

```
        if ( (currentnode == self._first) and (currentnode == self._last) and (previousnode == None) )
```

```
            ## list has only one element
```

```
            assert(self._first == self._last)
```

```
            self._first = None
```

```
            self._last = None
```

```
        elif (currentnode == self._first):
```

```
            ## first element being removed and list has more than 1 element
```

```
            assert(self._first.next != None)
```

```
            self._first = currentnode.next
```

```
        elif (currentnode == self._last):
```

```
            ## last element being removed and list has more than 1 element
```

```
            assert(self._first)
```

```
            previousnode.next = None
```

```
            self._last = previousnode
```

```
        else:
```

```
            ## You are removing middle element
```

```
            assert(self._first)
```

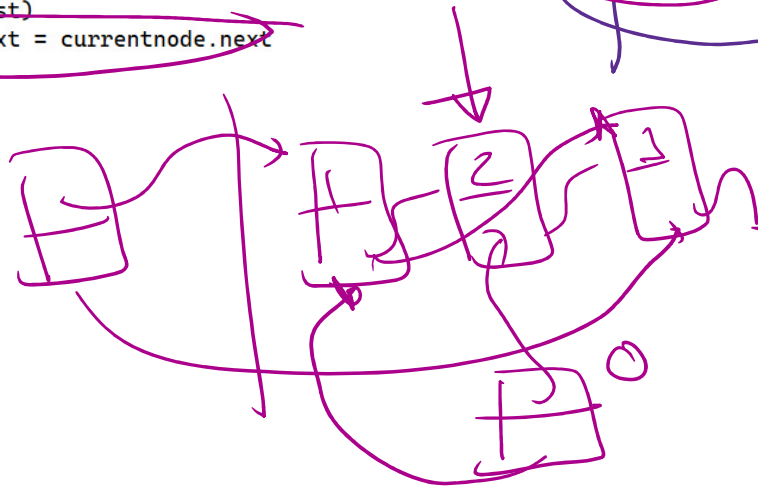
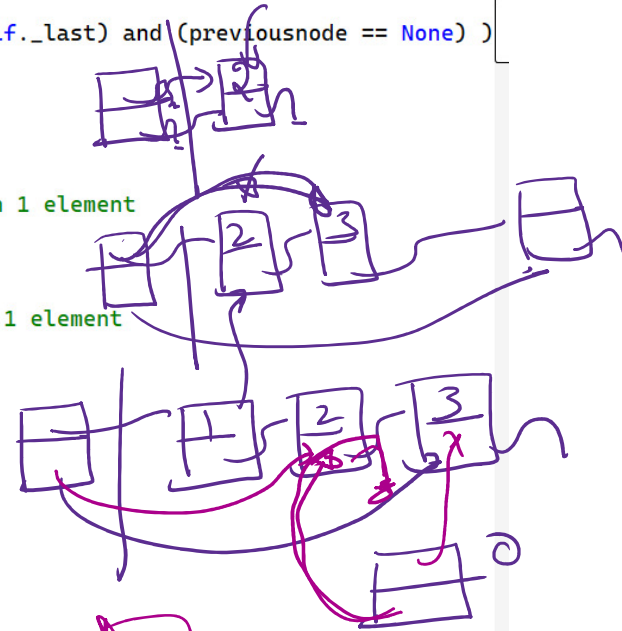
```
            assert(self._last)
```

```
            previousnode.next = currentnode.next
```

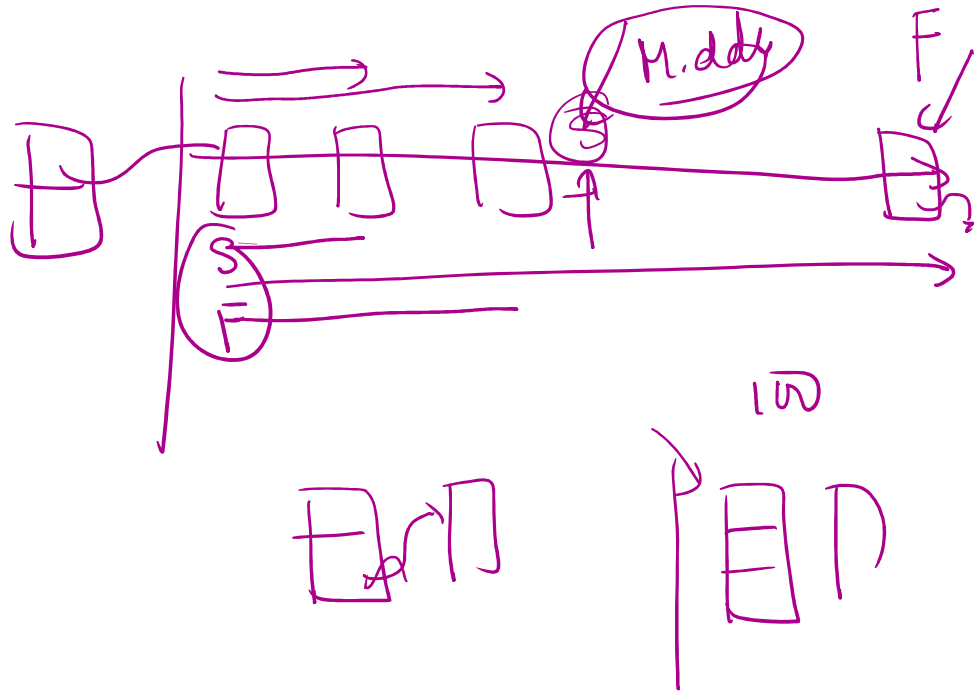
```
        return True
```

```
    else:
```

```
        return False
```



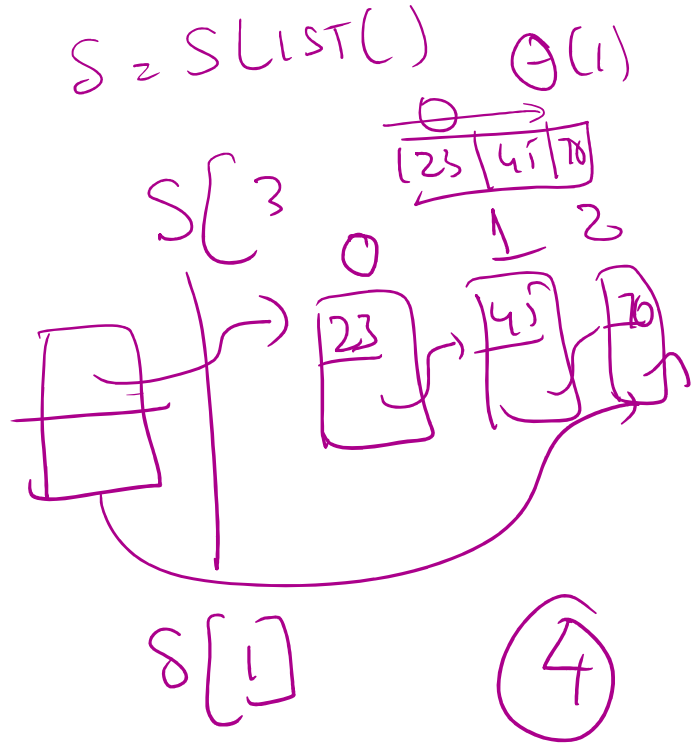
```
def find_mid_point(self)->'int':  
    if (self._first == None):  
        return 0  
    s = self._first  
    f = s.next  
    while(f != None):  
        f = f.next  
        if (f != None):  
            f = f.next  
            s = s.next  
    return s.val
```

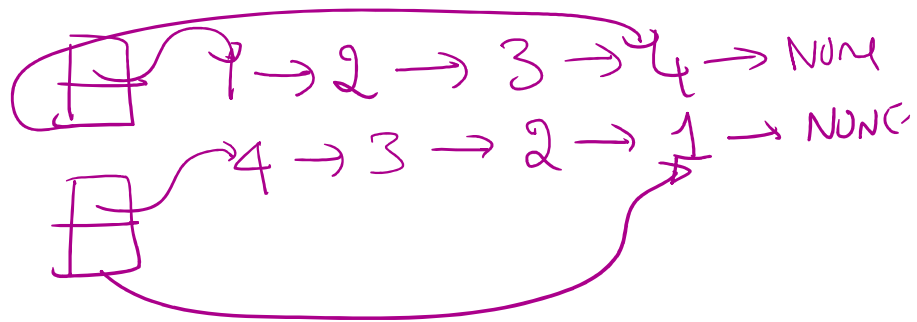


23

Wednesday, October 4, 2023 7:23 PM

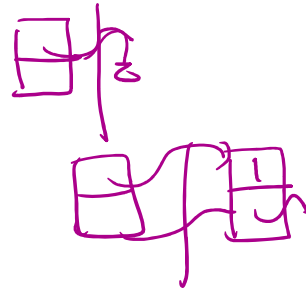
```
#####
def __getitem__(self,i:'int')->'int':
    n = self._first
    j = 0
    while(n != None):
        if (j == i):
            return n.val
        n = n.next ;
        j = j + 1
    return -1
```





Mayer

BASE



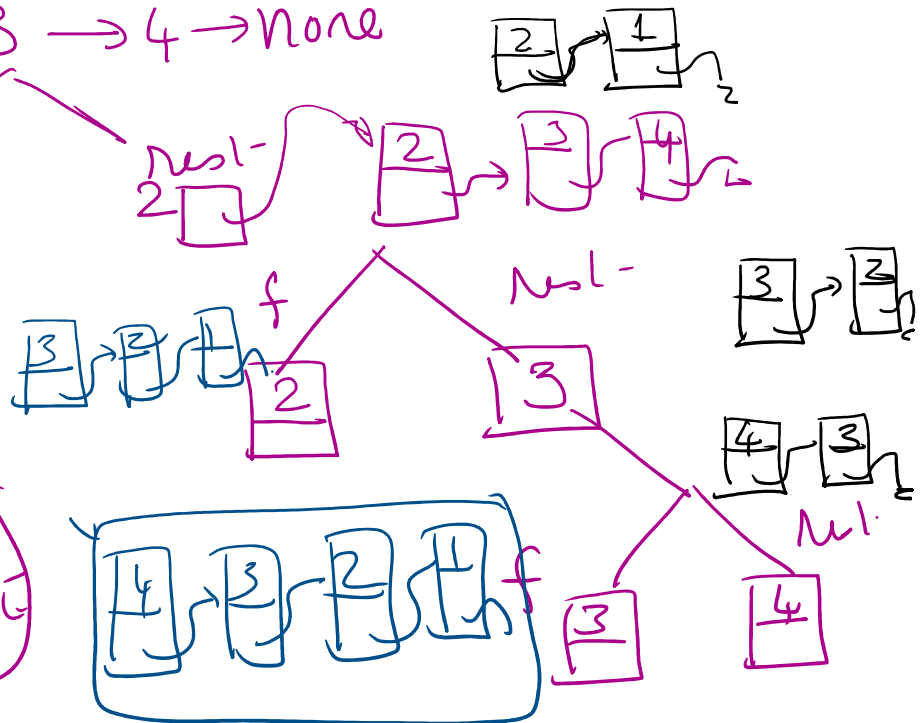
Divide & Conquer

f 1 → 2 → 3 → 4 → None

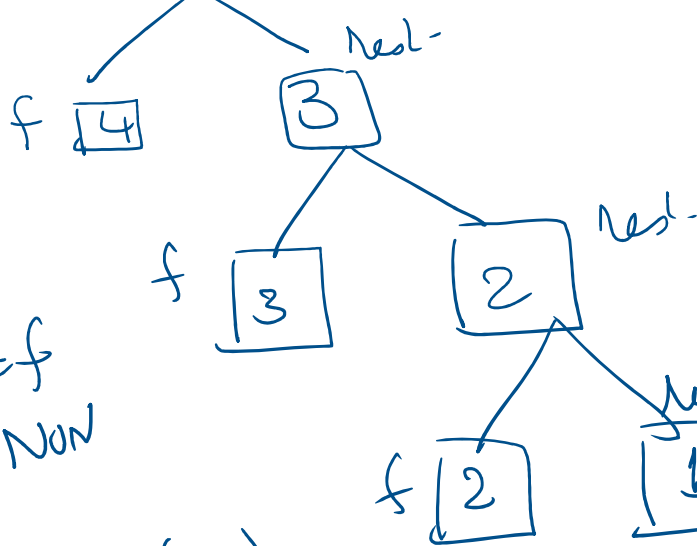
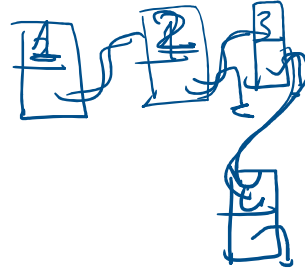
Conquer

Two Constant Operations

rest.next = f
f.next = None



f → 4 → 3 → 2 → 1 → None



ds
99
100

STACK OVERFLOW

Next.next = f
f.next = None

Time: $\Theta(N)$
SPA: $\Theta(N)$

```

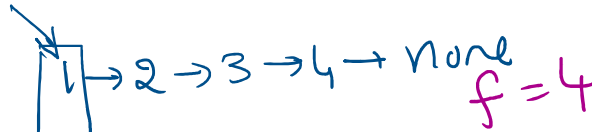
*****
def r_r(self, f: 'node'):
    ## Base case
    if (f.next == None):
        return
    ## Divide
    rest = f.next
    self.r_r(rest);
    ## Conquer
    f.next = None
    rest.next = f

```

```

def reverse_recur(self):
    f = self._first
    l = self._last
    if (f != None):
        self.r_r(f)
        self._last = f
        self._first = l

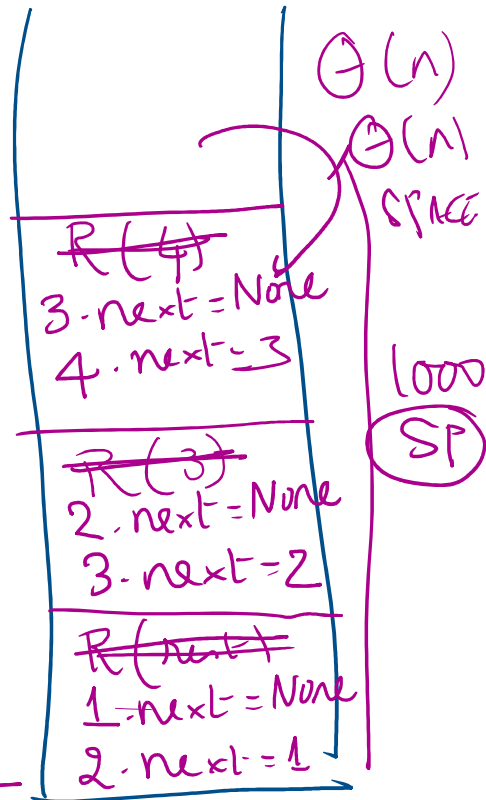
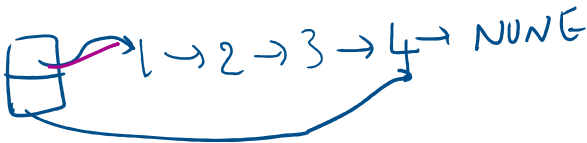
```



f = 3
next = 4

next = 2
next = 3

f = 1
next = 2



are given an array of n int
day 0 1 2 3 4
cost 5 10 4 6 12

Buy Day

2

sell day

4

Buy
8

8

W2

100\$

8

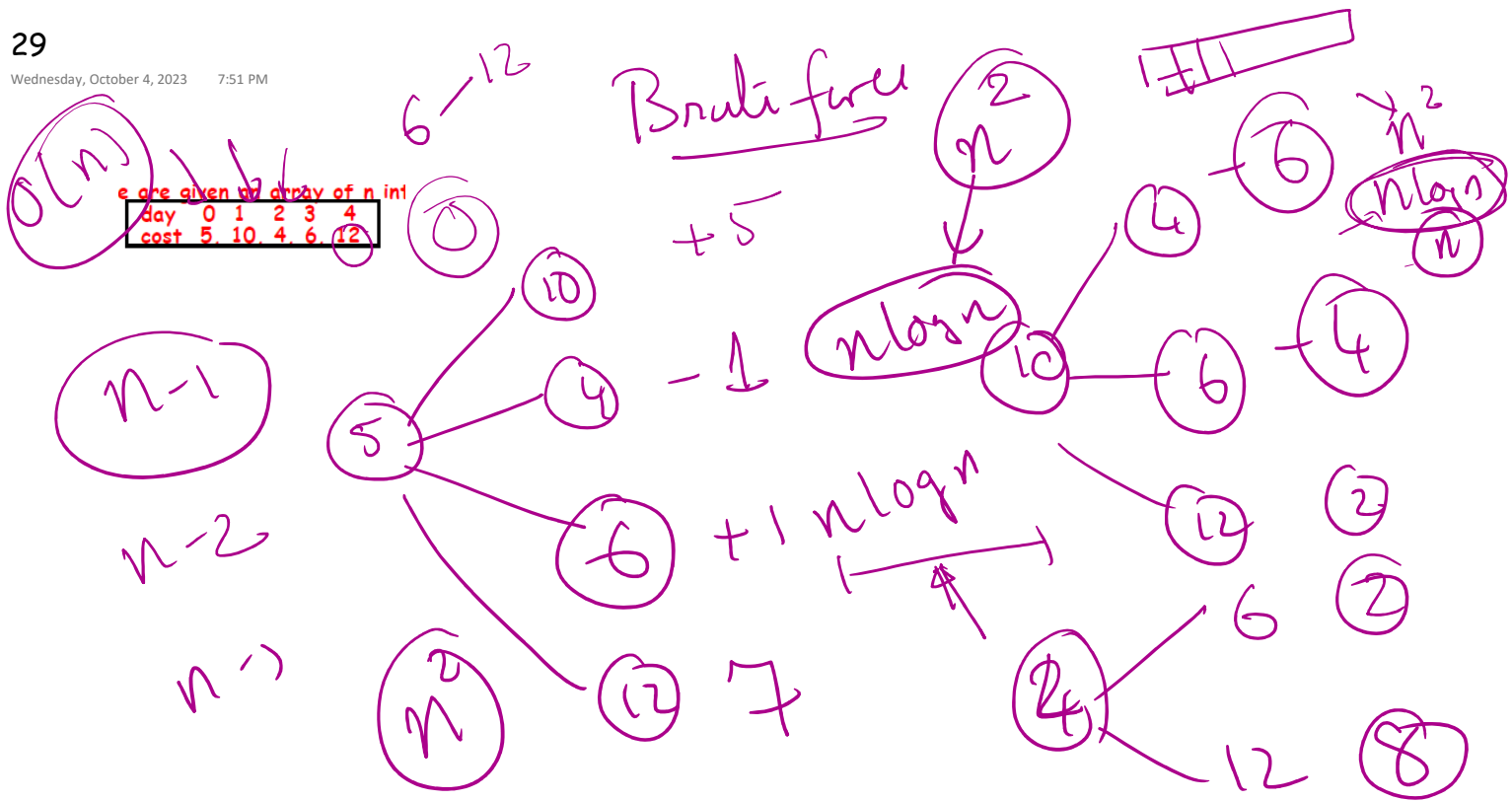
40%

101

2023

we are given an array of n int

day	0	1	2	3	4
cost	5	10	4	6	12

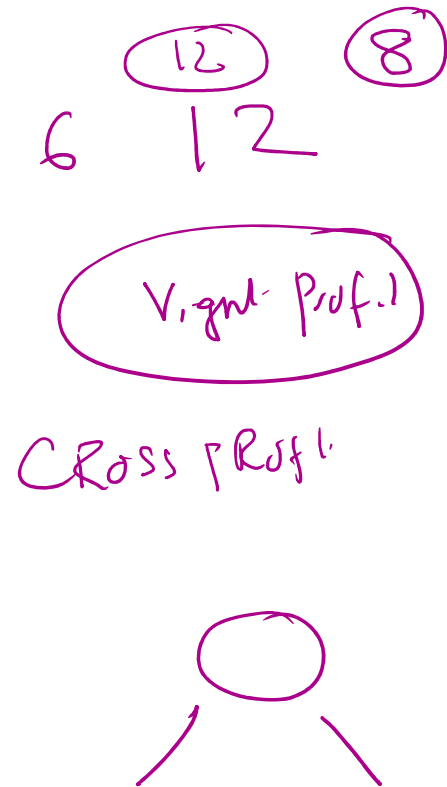
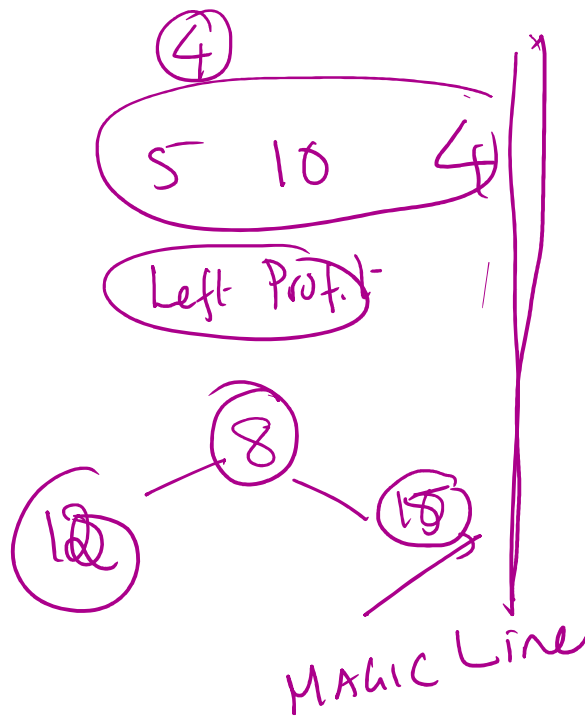


we are given an array of n int

day	0	1	2	3	4
cost	5	10	4	6	12



Buy \rightarrow Sell



31

we are given an array of n int

day	0	1	2	3	4
cost	5	10	4	6	12

cost 5, 10, 4, 6, 12

