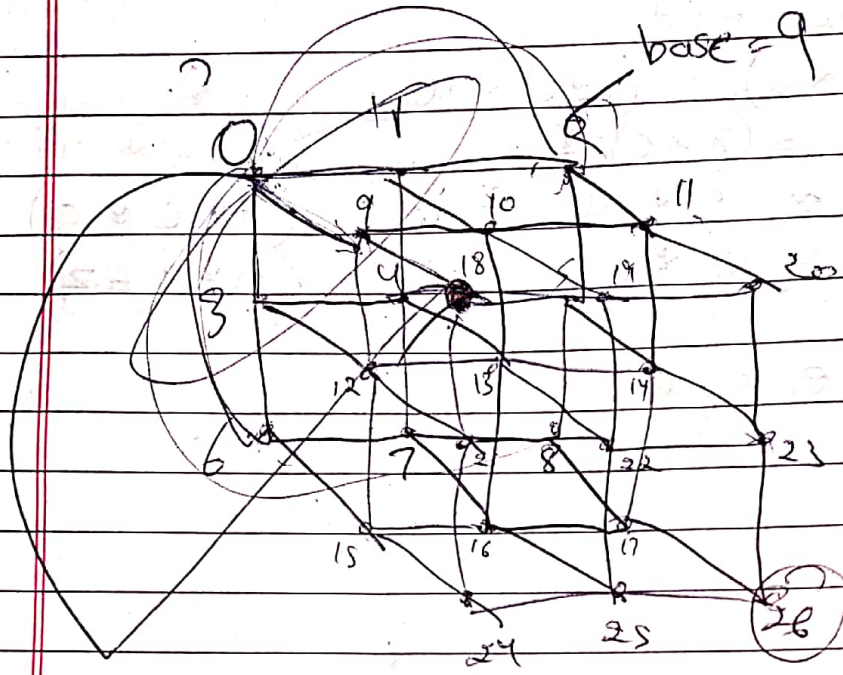


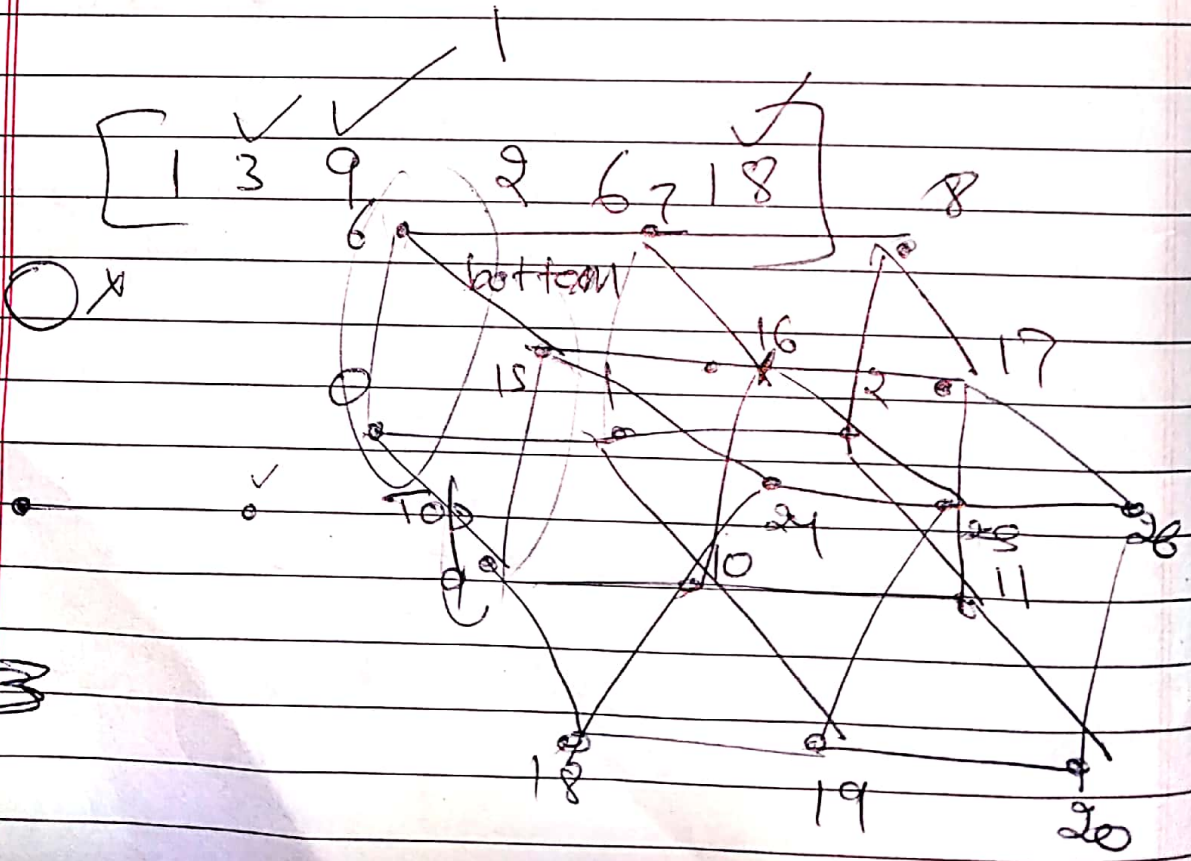
lower limit

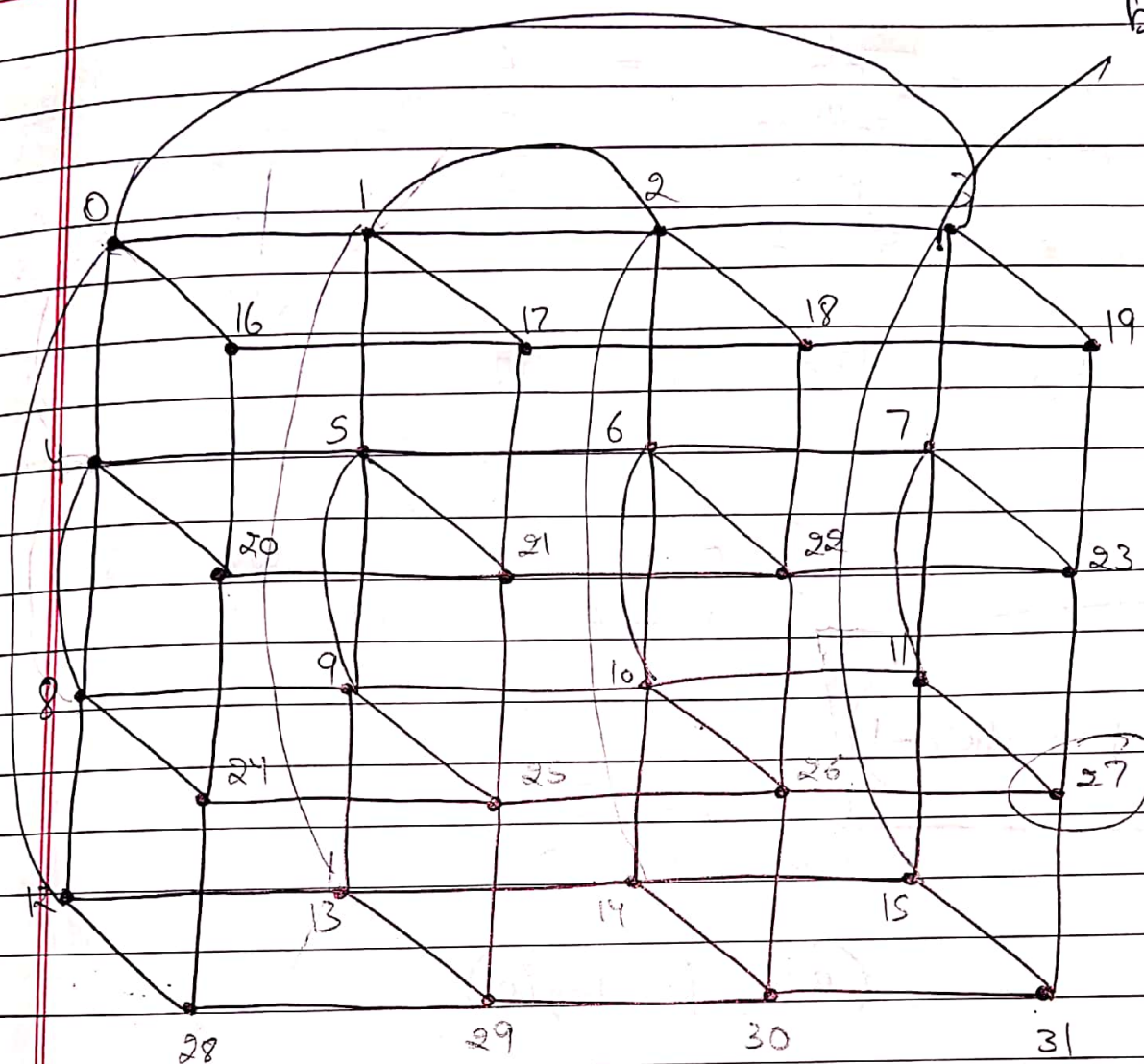
Date  
Page

4 numbers



2283





$0 + 9$

j, 18

$$9 + 9$$

layer =  $n \times n$

NUM \* NUM

$$0 + \log 2$$

$$2 \times \begin{bmatrix} 1, 5, 11 \\ 20 \end{bmatrix}$$

3, 12, 1, 4, 16



Top = 6  
bottom = 3 ✓  
left = 2  
right = 1 ✓  
back = 9 ✓  
front = 18

0 9 18

(27)

0 9  $i \neq 80$

number = 3

$j = 1, -6$

27

$i = (0 + \text{number} - 1)$

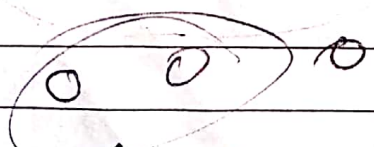
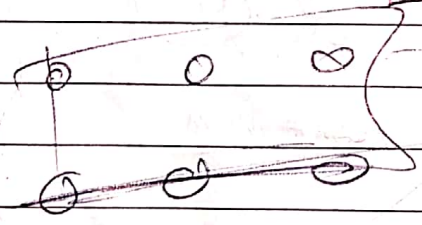
(acc, 2, 3)  
(acc, 4, 1)

(acc, j, logic)

if

for ( $i = 0; i < m; i++$  number \* number)

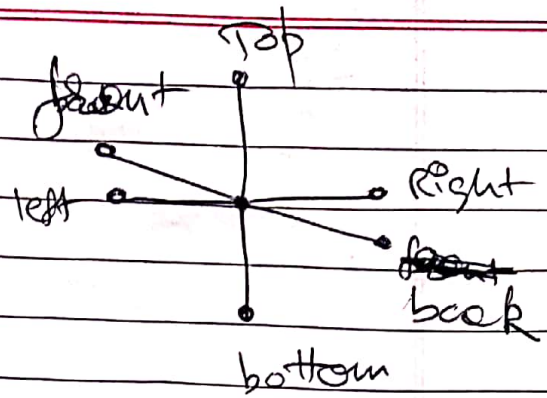
return last i



for ( $x = i; x < m; x++$  (number \* number))

{ last }

if ~~normal~~ - present do  
i-1



0 1 (2)

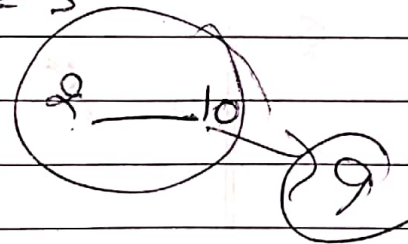
Last Level =  $\text{Math.floor}(n / \text{number} * \text{number})$

def recur (i, number, n) do  
if i > n do  
i - (number \* number)

i = 0..8

number = 3

n = 27



Enum.reduce (i, fn (x, acc) ->

if (acc > n) {

acc - layer

}  
else

acc +  $(\text{num} * \text{num})$  layer

if acc < n do

acc = acc + layer

x = acc

else

acc - layer

end  
acc  
end

front (i, layer, n)

i - layer

sum = i

enum-map (i..n, fn a →

if <sup>layer, i</sup> rem(~~layer~~) == 0 do

sum += layer

end  
end

1 2 3 4 5 6 7 8 9 10



Sum = 0  
number \* m

for (x = 0 ; x < m ; x += layer)

Sum += x ;

for ( x = 0 ; x < 20 ; x += 3 )  
Sum += x  
x = 7

0  
3  
6  
9  
12  
15  
18

Enum. \*

acc

Enum.reduce ( ~~acc~~ 0, fn (x, acc) →

if acc ≤ m do

~~acc~~

acc = acc + layer

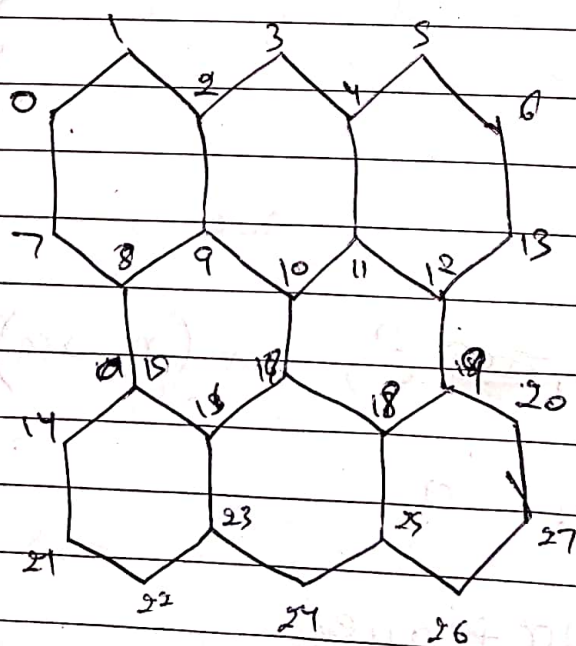
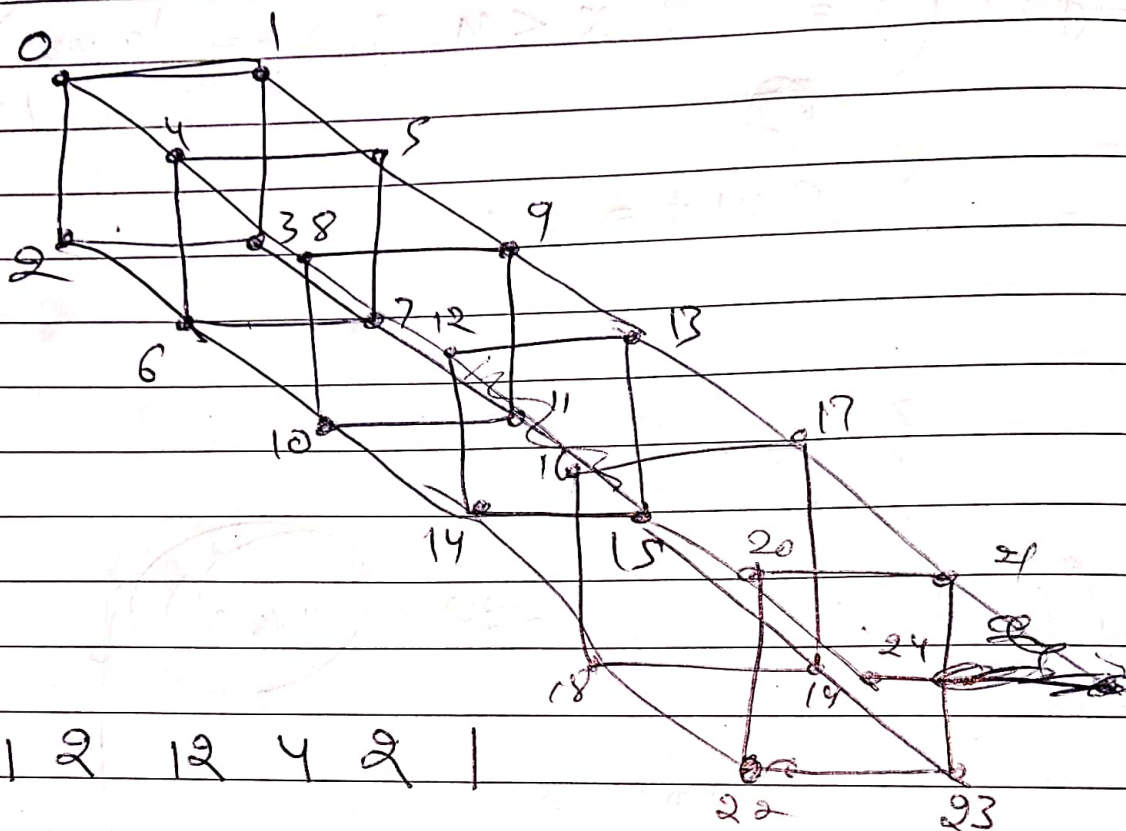
x = acc

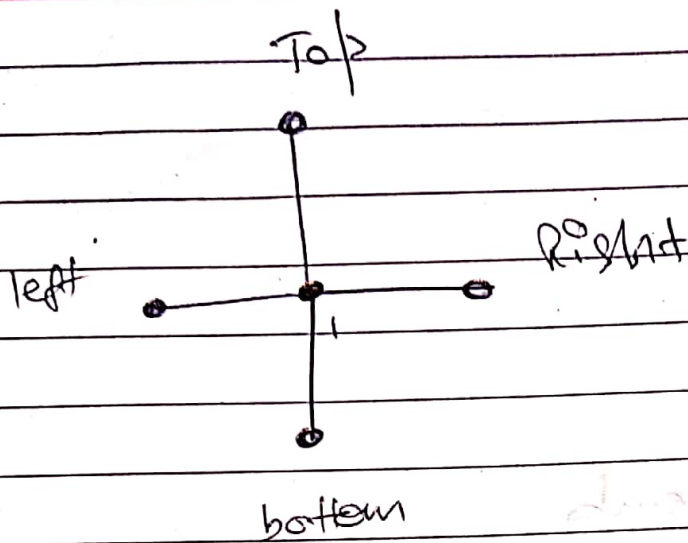
else

←

end

end





- 28
- ① = Top
  - ② = bottom
  - ③ = left
  - ④ = Right



$$j = 1$$



left

Right

bottom

