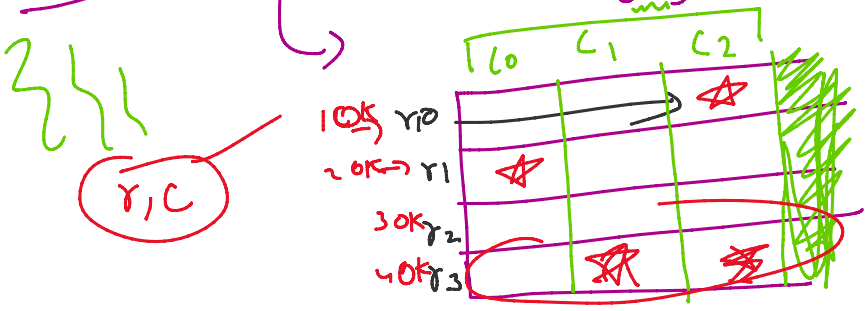
2D array

27 May 2023 12:20

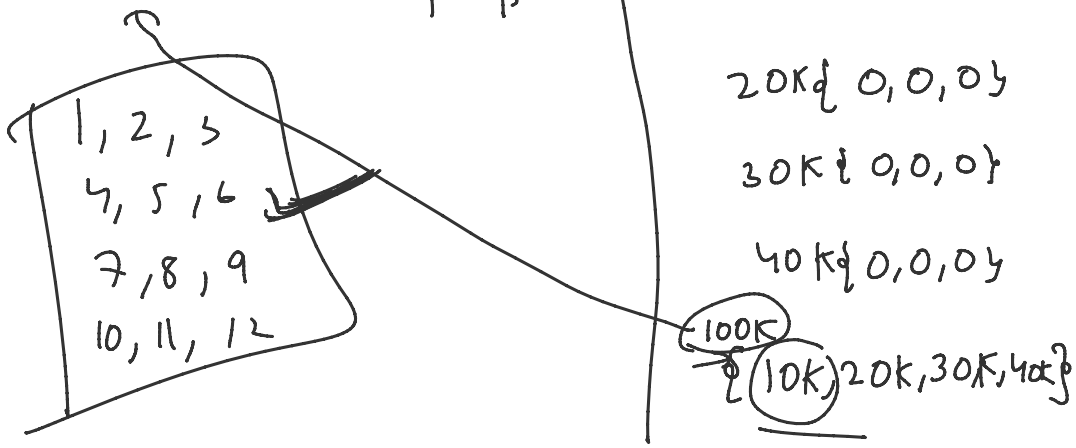
int [][] arr = new int [row] [col]

[4] [3]



{10K, 20K, 30K, 40K }

int [][] arr = new int [4] [3];

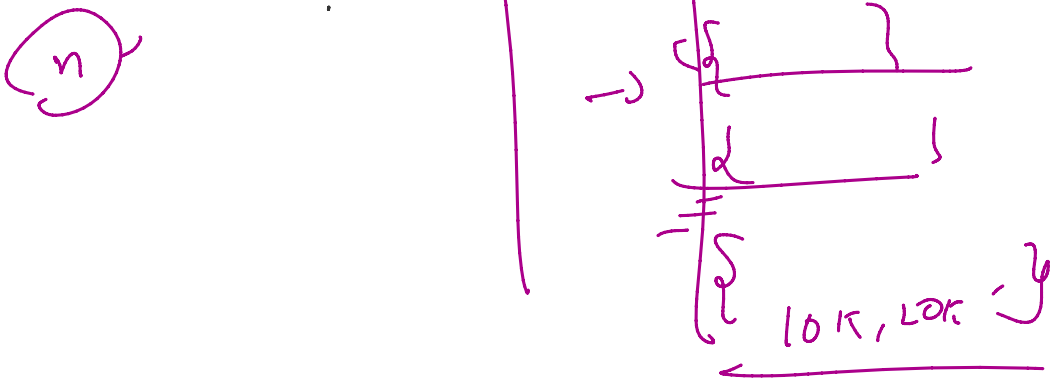


int [][] arr = new int [n] [m]

47:30

n x m

m + n - 1



Input

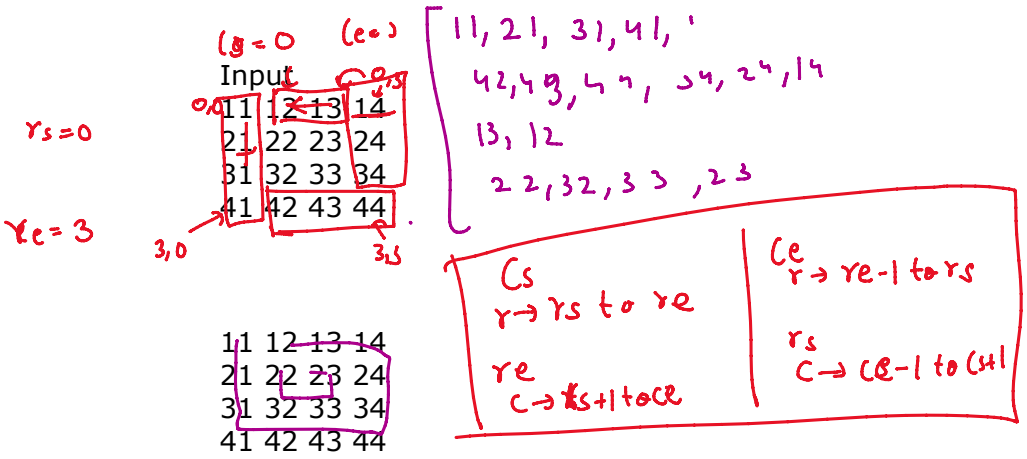
11 12 13 14
21 22 23 24
31 32 33 34
41 42 43 44

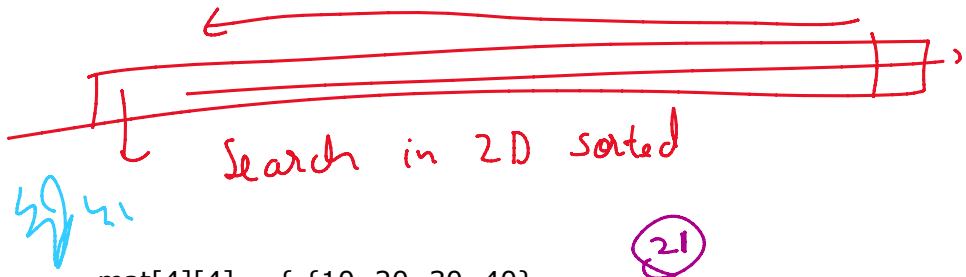
==========

Output

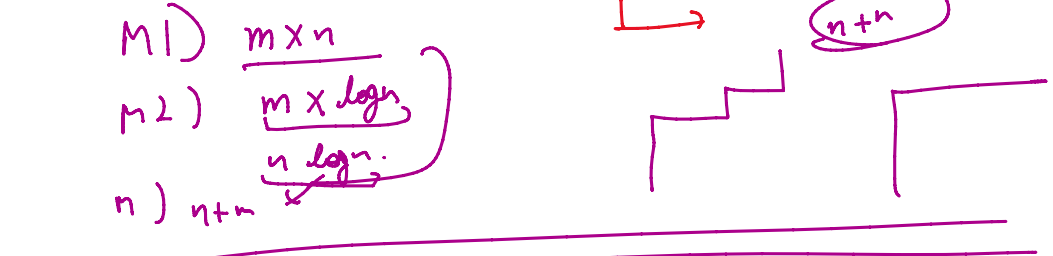
11 21 31 41 ,42 32 22 12 13 23 33 43 44 34 24 14

```
int[][] arr =
{{11,12,13,14} ,
{21 ,22 ,23, 24}, {31, 32 ,33 ,34} ,{41 ,42, 43, 44}
```






```
mat[4][4] = { {10, 20, 30, 40},
              {15, 25, 35, 45},
              {27, 28, 37, 48},
              {29, 33, 39, 50}};
```



Wrapper class

Primitive → non primitive

Primitive → non primitive

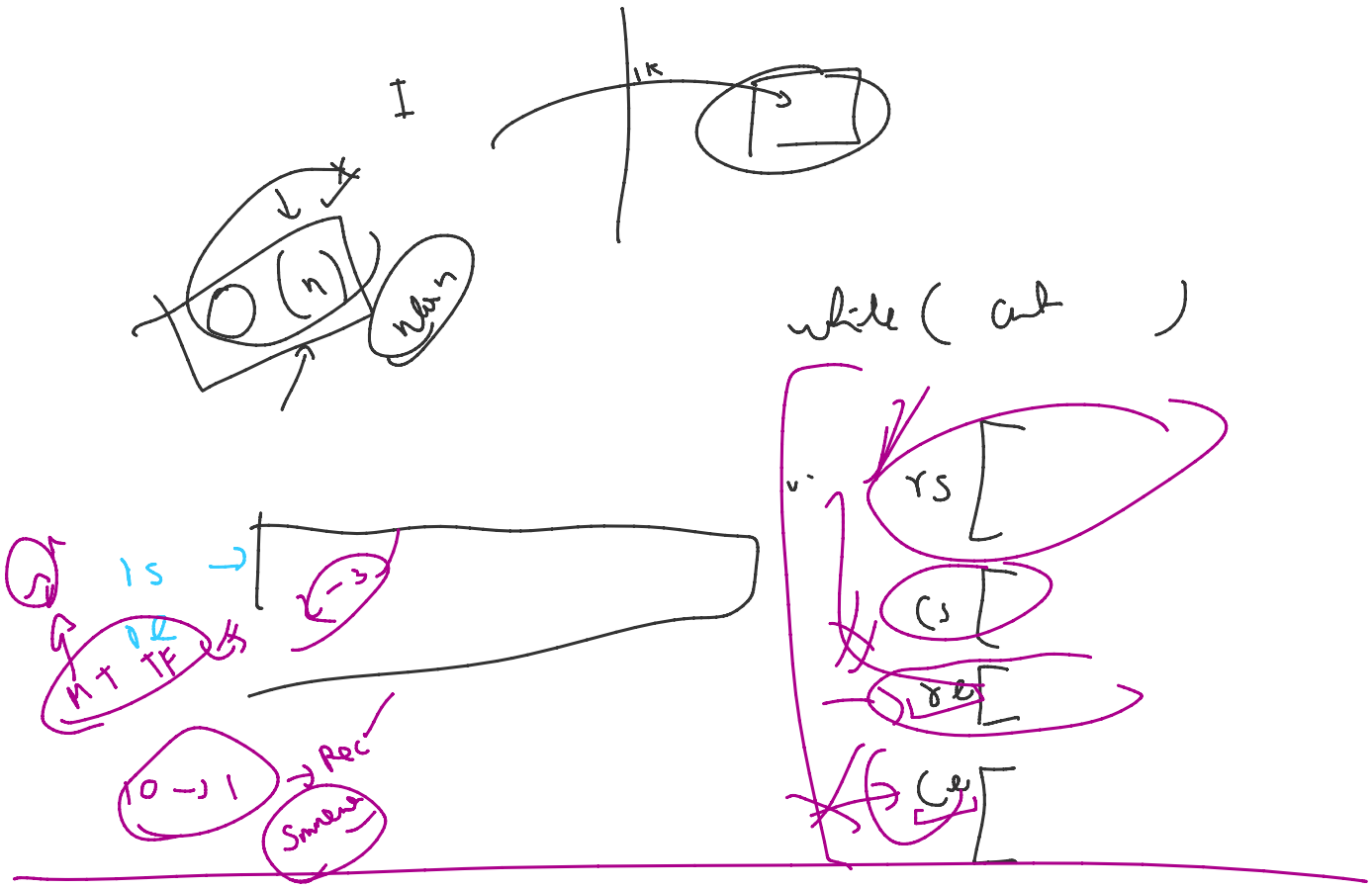
heaven → ↳ Wrapper class

primitive → stack → fast store

byte, short, int, long, float, double,
boolean, char

Wrapper class → Byte, Short, Integer, Long, Float, Double,
Boolean, Character

heap



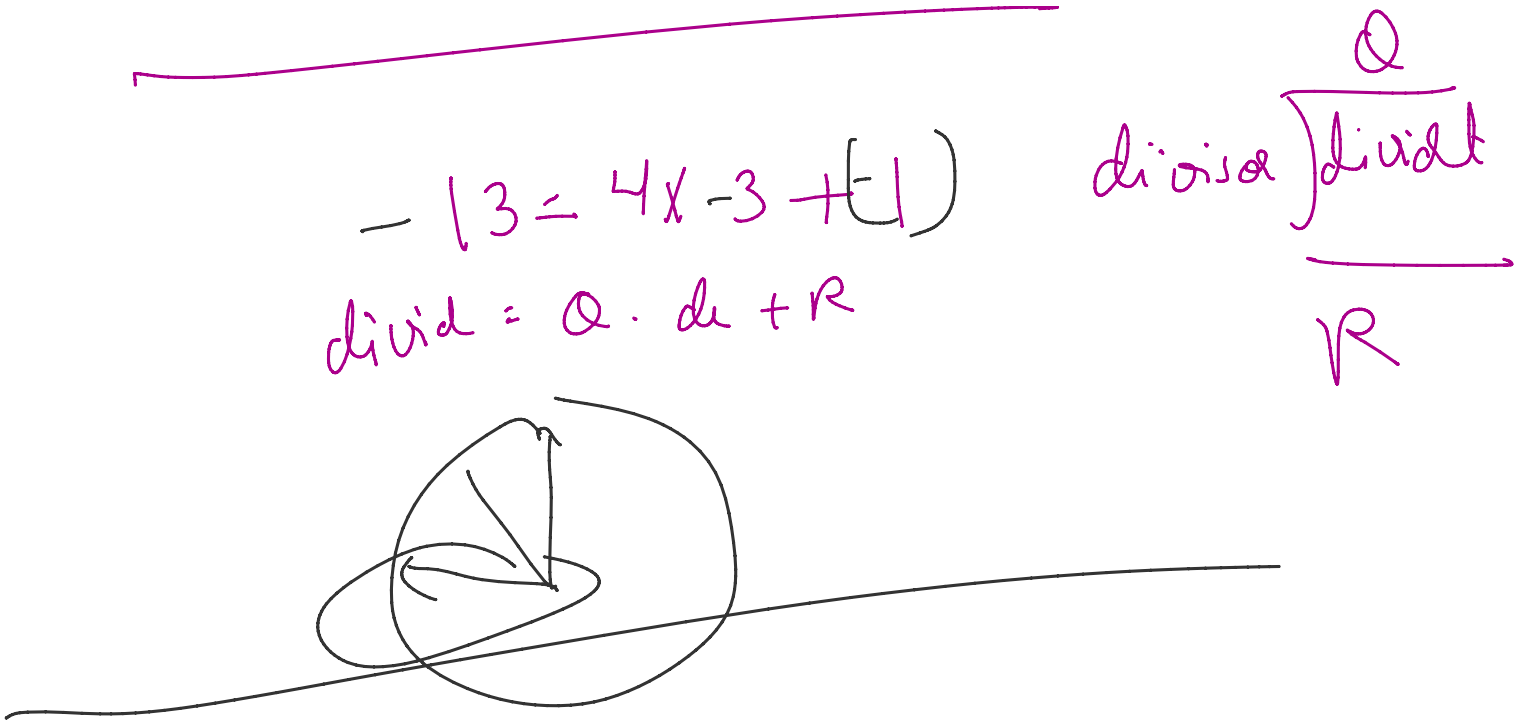
Climb stairs

BP → (0, n)

int sp1 = (1, n)

int sp2 = (2, n)

Q



The coding blocks members went to the success party of their first ever online boot-camp at Murthal. They ordered **P** number of paranthas. The stall has **L** cooks and each cook has a rank **R**. A cook with a rank **R** can cook 1 parantha in the first R minutes 1 more parantha in the next 2R minutes, 1 more parantha in 3R minutes and so on(he can only cook a complete parantha) (For example if a cook is ranked 2.. he will cook one parantha in 2 minutes one more parantha in the next 4 mins and one more in the next 6 minutes hence in total 12 minutes he cooks 3 paranthas. In 13 minutes also he can cook only 3 paranthas as he does not have enough time for the 4th parantha). Calculate the minimum time needed to cook all the paranthas.

Input Format

First line contains P, the number of pratha ordered. In the next line the first integer denotes the number of cooks L and L integers follow in the Next line each denoting the rank of a cook.

Output Format

Print an integer which tells the number of minutes needed to get the order done.

Sample Input

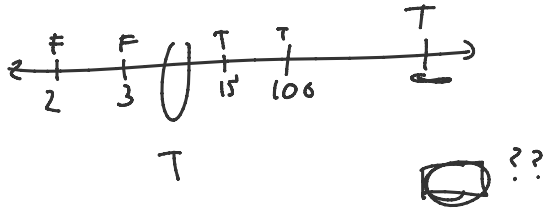
10
4
1 2 3 4

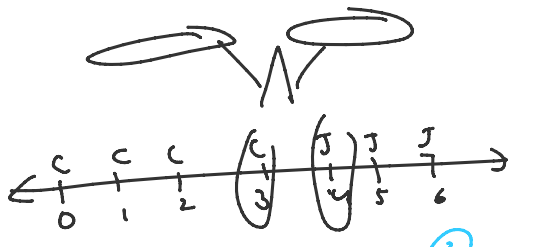
Sample Output

12

Explanation

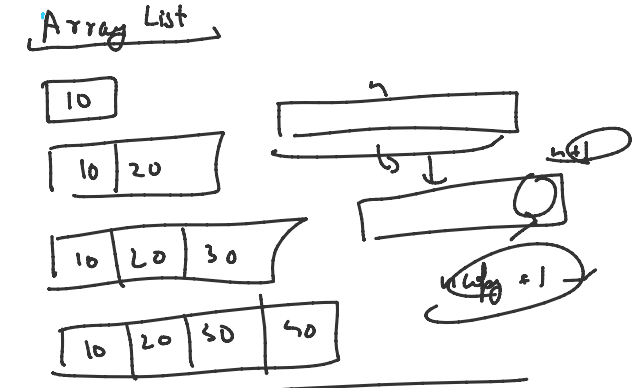
First cook with rank 1 cooks 4 paranthas in 10 minutes (1+2+3+4).
Second cook with rank 2 cooks 3 paranthas in 12 minutes (2+4+6)
Third cook with rank 3 cooks 2 paranthas in 9 minutes (3+6) Fourth cook with rank 4 only needs to cook one last remaining parantha. He can do that in 4 minutes.
Since these cooks cook parallely, the total time taken will be the maximum of the four i.e. 12 minutes.





{ 10, 50, 12, 16 } ✓ 3

{ 10, 50, 12, 16 }



Array List → Dynamic Array

int[] arr = new int[3];

[ ] → [10] → size = 1

[10,20] → size = 2
0 1

[10,20,30]

{ 10, 20, 30, 40, 50 }
0 1 2 3 4

2 sorted arrays. find elements which are present in both arrays

- 5,7,10,10,10,20,30,30,50,60,60,80
- 10,10,15,20,30,30,30,60,70,80,80,90

10,10,20,30,30,60,80

- 5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,7,10,10,10,20,30,30,50,60,60,80
- 10,10,15,20,30,30,30,60,70,80,80,90

10, 10, 20, 30, 30, 60, 80

- 5,7,10,10,10,20,30,30,50,60,60,80
- 10,10,15,20,30,30,30,60,70,80,80,90

10, 10, 20, 30, 30, 60, 80

one: [9,8,7,5]
two: [9,6,7]
[1,0,8,4,2]

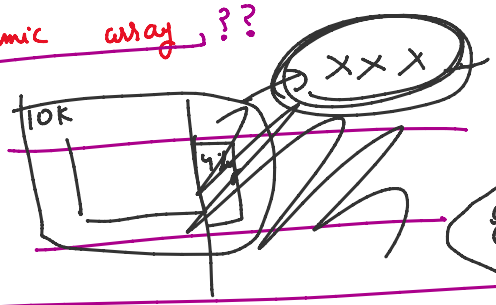
3min

{9,9,9,0,1,1,9}
{1,0}
A.L
4,0,2,2,9

```
int[] one = { 9, 8, 7, 5};
int[] two = { 9, 6, 7 };
```

{2,4,8,0}
1

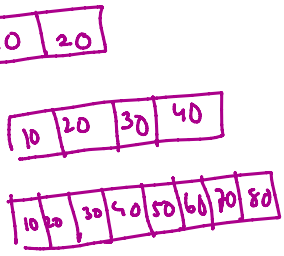
Q dynamic array ??

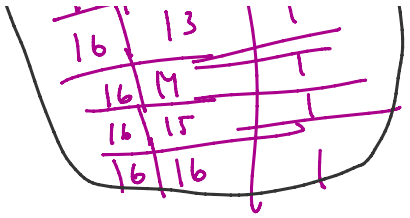


grow ×2
initial capacity = ??

cap	size	operations
2	0	
2	1	1
2	2	1
4	3	2+1
4	4	1
8	5	4+1
8	6	1
8	7	1
8	8	1
16	9	8+1
16	10	1
16	11	1
16	12	1
16	13	1
16	M	1

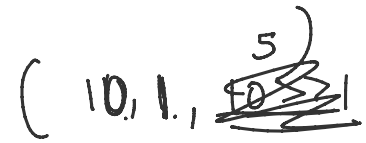
add(10)
add(20)
add(30)
40
50
60
2
16
add





min cap = 1

old = 10



1010 >>1

101

5

×1.5

1
10
100
1000

Complexity

A

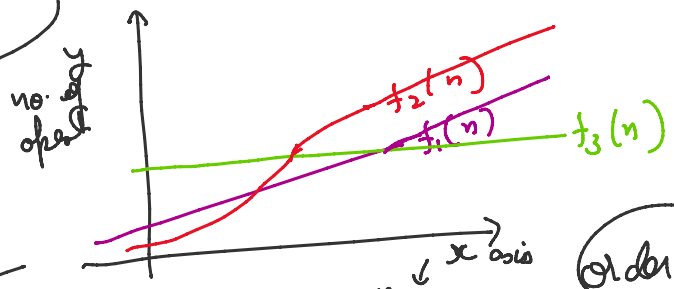
10hour

100

B

1min

no. of operⁿ's:



$t_3(n)$ $10^6 + \log n$

$f(2) = n^2 + 5n - 500$

$t_1(n) = 2n + 7$

$f(n)$ → large values of n worst case

Big O → O

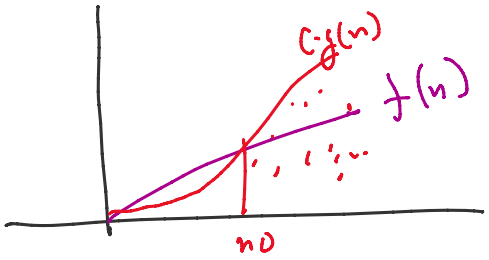
Big O

$c \cdot g(n) \geq f(n) \quad \forall n \geq n_0$

$O(g(n))$

↳ large values of n

$f(n) \xrightarrow{\text{converge}}$ worst case



$f(n) = 5n + 7 \rightarrow O(n)$

$5n + 7 \leq 12n \quad \forall n \geq 2$

$f(n) = 2n^2 + 5n - 500 \leq 2n^2 + 5n^2 + 500n^2 \quad \forall n \geq 100$

$\leq 510n^2$

$O(n^2)$

$f(n) = a_x n^x + a_{x-1} n^{x-1} + a_{x-2} n^{x-2} + \dots + a_0 n^0$

↳ $O(n^x)$

$a_x n^x + a_{x-1} n^x + a_{x-2} n^x + \dots + a_0 n^x$

$n^x \quad k \quad \geq f(n) \quad n \geq n_{00}$

$O(kn^x)$

→ Maths context

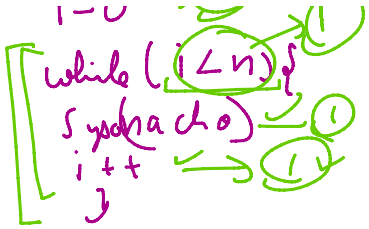
→ Variable context

→ Point context

$i = 0 \rightarrow 1$

while ($i < n$) { $\rightarrow 1$

$f(n) = 1 + 3n$



$f(n) = 1+3(k)$

$f(n) = 1+3n$ → $O(n)$

```
i = 1
while (i<n){
  sysout(??
  i = i * 2;
}
```

$f(k) = 1+k$ → $O(k)$