The code follow PEP 8 guidelines.

Output:

```
C:\Users\Krish\AppData\Local\Programs\Python\Python312\python.exe "C:\Users\Krish\Python\Test\DAA LAB 4.py
O iversions occur for 1 times
1 iversions occur for 3 times
2 iversions occur for 10 times
3 iversions occur for 12 times
4 iversions occur for 19 times
5 iversions occur for 13 times
6 iversions occur for 15 times
7 iversions occur for 14 times
8 iversions occur for 10 times
9 iversions occur for 3 times
10 iversions occur for 0 times
C:\Users\Krish\Python\Test\.venv\Scripts\python.exe "C:\Users\Krish\Python\Test\DAA LAB 4.py"
Number of inversions: 10
Number of inversions: 6
Number of inversions: 1
Number of inversions: 0
Number of inversions: 8
Number of inversions: The array is not compatible!!
C:\Users\Krish\Python\Test\.venv\Scripts\python.exe "C:\Users\Krish\Python\Test\DAA LAB 4.py"
35937749302050
706776567696
69192562487288
959702547700
1753874300073288
Invalid Datatype
Invalid Datatype
Invalid Datatype
Invalid Datatype
Invalid Datatype
Process finished with exit code 0
```

```
C:\Users\Krish\Python\Test\.venv\Scripts\python.exe "C:\Users\Krish\Python\Test\DAA LAB 4.py"
35937749302050
706776567696
69192562487288
959702547700
1753874300073288
Invalid Datatype
```

Conclusion:

This experiment shows the power of using divide and conquer approach over the brute force approach. The count inversion algorithm has time complexity of O(nlogn) same as merge sort because it leverages merge sort with just counting the number of inversion as additional feature. The brute force integer multiplication has time complexity of $O(n^2)$ and divide and conquer lowers it to $O(n^{\log 3})$. The number of inversions follows a normal distribution or a bell curve.

Date page # Counting Inversions # Course choice arrays of students Count (L,R): i,j <-0,0 n,n2<- len(L), len(R) count <-0 ans <- [7 while i = n1 or j!= n2 do: y L[i] > R[j] then add R[j] to ans i++
count ~ ni-i+ count else add L[i] to answer: ý 1/== n1 then 1/2 do while j!= n2 do add REiJ to ans ij = n2 then while i!=n1 do add LEIJ to ans secturer lasses counts

Classmate [1,2,3,4] not compatible not compatible [6,2,1,4] 9 ® [1,2,3,4,5,6] - not compatible 9[7,8,9,10] > not compatible (0)- not compatible

5



classmate

Time complicity:

Court Inversions:

T(n) = 2T(n|2) + 0

2 subportion marge of of anoth 12 sorted array and counting

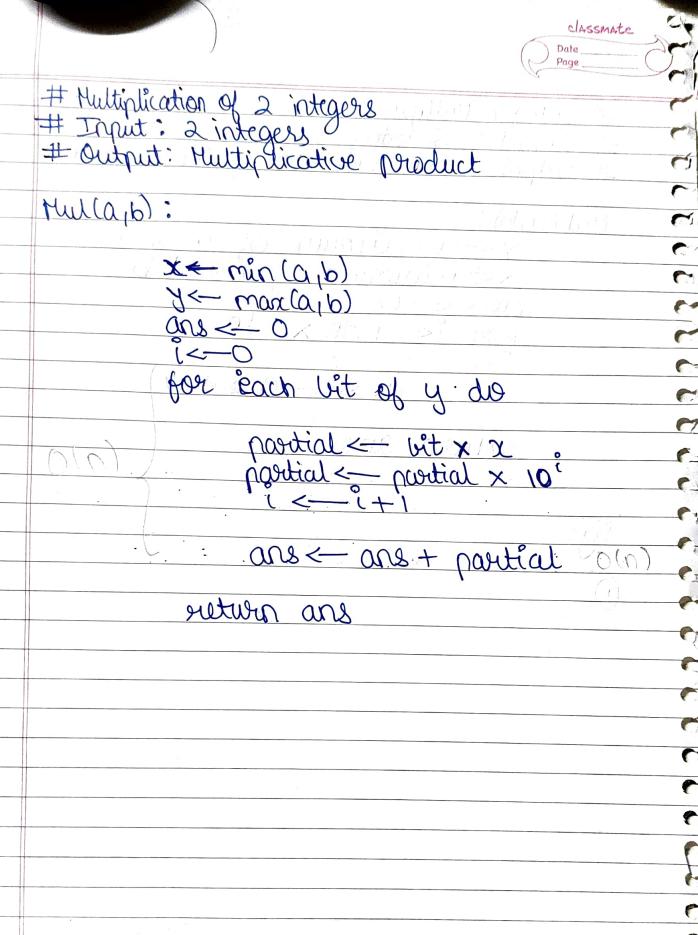
:. $a=2,b=2, \pm (n)=nd=n' \Rightarrow d=1$

by master theorem,

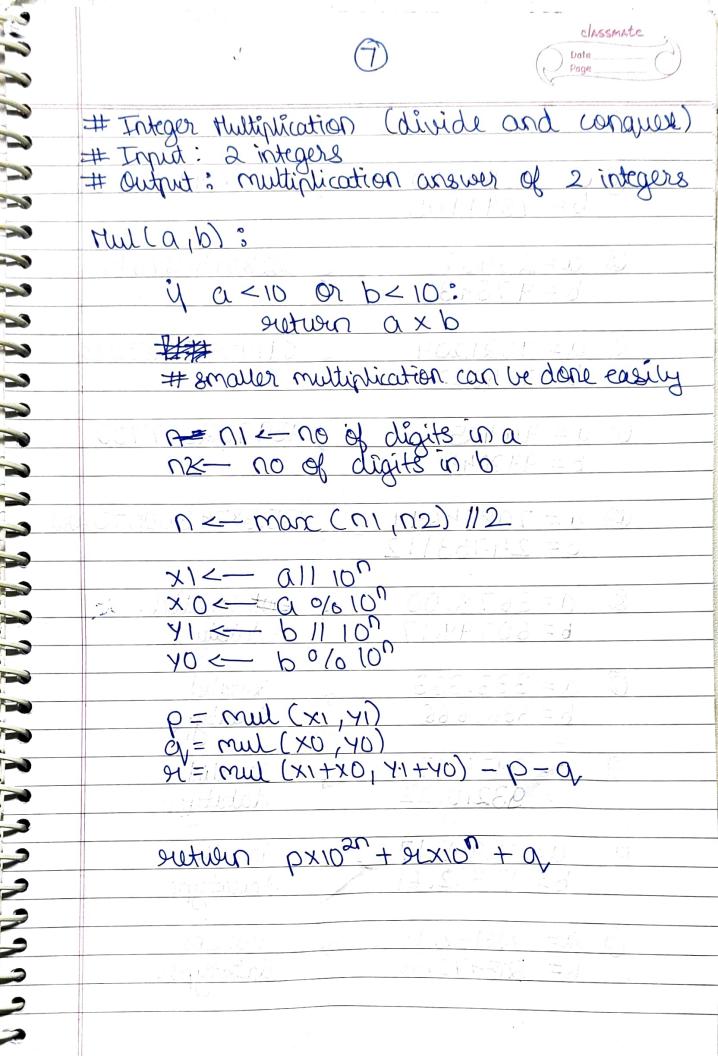
bd = 2 = 2 = a

:. Time complexity = Oladloga)

= O(U/ddu)



C



Classmate

Test Case:

35937749302050 (i) a= 4554210 b= 7891105

228814887696 0 a=234578 b = 975432

3 a= 9821204 69192562487288 6 = 70 45 222

(h) a = 966425 959702547700 b = 993044

© a= 70854699 175 3874 300073288 6= 24753112

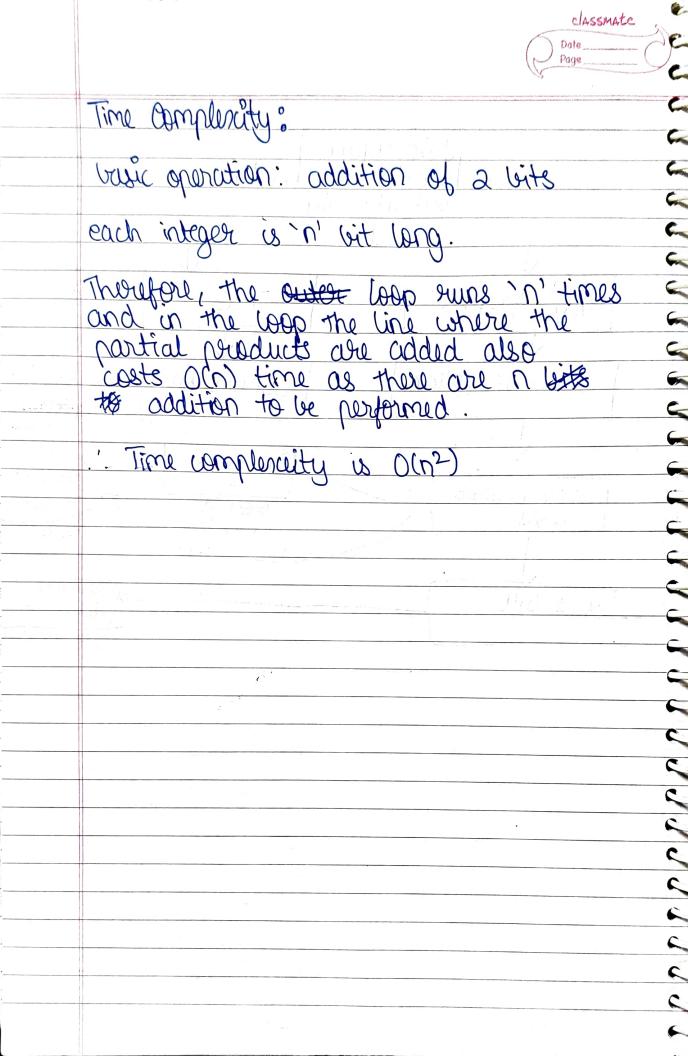
a = 5678.00pot invalid datatype b=60799.17

0 $\alpha = 333.333$ 'muralid datatype b = 666.666

8 a=7974017 bibouri b= 93210.22 datatypo

(0) billeri a= 12345.17 dutatype b= 54312.61

(1) a= 786132.14 invalid 67 215473.66 datatype.



Time complicity:

Karatsulza Agorithm

T(n) = 3T(n/2) + 0000 n

3 subpravens addition of PIONIAL

T(n) = aT(n|b) + f(n) $a = 3, b = 2, f(n) = n' = n^d$ of a = 1

by master theorem,

& Time complexity = O(n log 23)

[: bd = 21 = 2 < a]