The first array is always the smaller array.

This is based on binary search

![A picture containing text, black

Description automatically generated]()

See here total size is 10 so 5 we need in first half

1247 and 2 here 7 is more than 3 of next array hence I for first array nad j for 2nd array

Do i--, and j++ we get sorted half

See 2nd example13 and 23 6 here 6>4 hence i++ and j—till we get half

![A picture containing diagram

Description automatically generated]()

Here low is 0 min element

High is max element of less size array

Take 2 variable

Cut1 for lower size array =low+high/2=2 here call it

Cut2 for higher size=10/2 =5-(low+high)/2=3 here

Also take variable l1->at lower array at 1 index=12, l2=higher array at 2nd pos 3, r1 at 2nd index 14, r2=a higher array at 3 index 4 elements. SEE below.

![Graphical user interface, application, icon

Description automatically generated]()

Since 12 is greater than 4 do mid-1 high=1 then from 4.

![A picture containing diagram

Description automatically generated]()

This is binary search as mid earlier was 2 sol pass(low,mid-1) see that how we got cut1 and 2

Here 9 greater than 7 hence pass(mid+1,high)

![Diagram, schematic

Description automatically generated]()

All sorted now.

Here for 7,4 take maximum and 12,4 take min and divide by 2.

Text

Description automatically generated

Here l1 if empty or cut1 0 take INT\_MIN and r1 same cut2 0 take INT\_MAX.same for l1 and r2.

![Text

Description automatically generated]()

Sol:

![Graphical user interface, text, application

Description automatically generated]()