Ques Minimize Product

num1	2	10	5	7
num2	9	1	7	4

} Product Sum

$10\times9 + 2\times1 + 5\times7 + 7\times4$

$90 + 2 + 35 + 28$

155

$2\times9 + 10\times1 + 5\times7 + 7\times4$

$18 + 10 + 35 + 28$

91

num1	2	5	7	10
num2	9	7	4	1

$2\times1 + 5\times4 + 7\times7 + 10\times9$

$2 + 20 + 49 + 90$

161

$2\times9 + 5\times7 + 7\times4 + 10\times1$

$18 + 35 + 28 + 10$

91

1 10

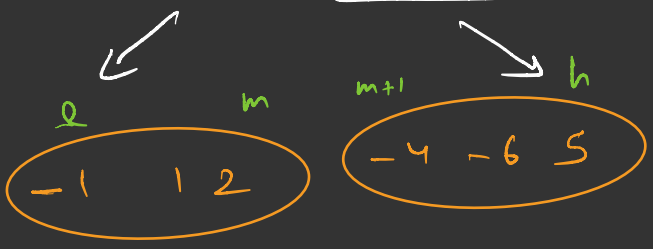
2 20 ⇒ 202

20 + 20 ⇒ 40

Ques Separate positive and negative

1 -1 2 -4 5 -6 → sorting -6 -4 -1 1 2 5

-1 -4 -6 1 2 5



-1 -4 -6 1 2 5

-1	-4	-6	1	2	5

TC ⇒ n Log n

```
merge(arr, l, m, h) {
  aux  k=0
  for(i=l; i<=m; i++) {
    if (arr[i]<0) {
      aux[k++] = arr[i];
    }
  }
  for(i=m+1; i<=h; i++) {
    if (arr[i]<0) {
      aux[k++] = arr[i];
    }
  }
  for(i=l; i<=m; i++) {
    if (arr[i]>0) {
      aux[k++] = arr[i];
    }
  }
  for(i=m+1; i<=h; i++) {
    if (arr[i]>0) {
      aux[k++] = arr[i];
    }
  }
}
```

1 -10 -4 3 7 9 -1 6

-10 -4 -1 1 3 7 9 6

Ques Inversion of array

2 4 1 3 5 ⇒ 1 + 2 = 3

1 2 3 4 5

5 4 3 2 1 ⇒ 4 + 3 + 2 + 1

⇒ 10

```
for(i=0; i<n; i++) {
  for(j=i+1 j<n; j++)
      if arr[j] < arr[i]
          count ++

return count
```

Expectation

mergeSort(arr, 0, 4)

⇒ 3

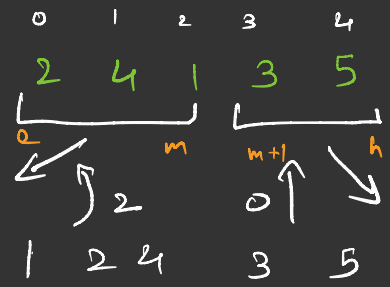
Faith

sort half array and return inversion count

m = l + h / 2

⇒ 0 + 4 / 2

⇒ 2



inv [1st part]

+ inv [2nd part]

+ inv [1st over 2nd]

arr[i] > arr[j]

m - i + 1

2 3 4 5

i m

m - i ⇒ 5 - 2 ⇒ 3

+1