Brute:

def numberOfInversions(a : List[int], n : int) -> int:

    count=0

    for i in range(1,n):

        for j in range(0,i):

            if a[j]>a[i]:

                count+=1

    return count

Optimal:

from typing import \*

def merge(arr,s,e,mid):

    cnt=0

    arr3=[]

    i=s

    j=mid+1

    while(i<=mid and j<=e):

        if arr[i]<=arr[j]:

            arr3.append(arr[i])

            i+=1

        else:

            cnt+=mid-i+1#if this no. is greater then then left of the elements are too since they are sorted

            arr3.append(arr[j])

            j+=1

    while(i<=mid):

        arr3.append(arr[i])

        i+=1

    while(j<=e):

        arr3.append(arr[j])

        j+=1

    for i in range(s,e+1):

        arr[i]=arr3[i-s]

    return cnt

def mergesort(a,s,e):

    cnt=0

    if s>=e:

        return cnt

    mid=(s+e)//2

    cnt+=mergesort(a,s,mid)

    cnt+=mergesort(a,mid+1,e)

    cnt+=merge(a,s,e,mid)

    return cnt

def numberOfInversions(a : List[int], n : int) -> int:

    count=mergesort(a,0,n-1)

    return count