Jesse Yang (23307563), Qichong Huang (23311085)

Fines　Explanation

Explanation

The Fines task can first be briefly concluded as a ***sorting problem***. In Fines, we are asked to count that in a given list, ***the total numbers of disorder integers,*** which should be in ***descending order***.

For example, countFines({ 4, 9, 7, 2, 1, 3 }) should return integer 4.

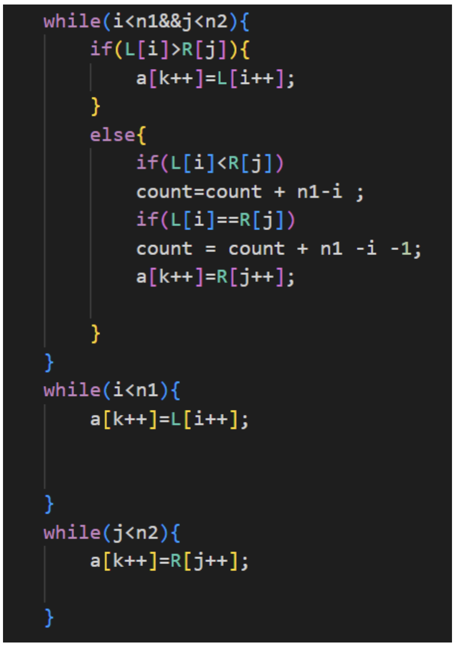
Counting the disorder integers in the list, we are, in fact sorting the list in descending order. In other words, if we implement a sorting algorithm to the solution, we can get the result we want by keep counting the disorder integer while sorting takes place.

There are many options. But if we want to pass the large test cases, we need a sorting algorithm whose computational complexity is better than O(N^2).

From what we have learned in this course, ***Merge Sort***, will be perfect. （computational complexity is O(NlogN))

Since we have tried to write a Merge Sort, implementing this algorithm to the solution will be easy. We only need to change few places to make it sort in descending order.

The key part is this.



```
while(i<n1&&j<n2){
    if(L[i]>R[j]){
        a[k++]=L[i++];
    }
    else{
        if(L[i]<R[j])
        count=count + n1-i ;
        if(L[i]==R[j])
        count = count + n1 -i -1;
        a[k++]=R[j++];

    }
}
while(i<n1){
    a[k++]=L[i++];


}
while(j<n2){
    a[k++]=R[j++];

}
```

If we ignore the variable count, this is just a part of descending Merge Sort.

As mention above we need to count the disorder integers in the middle of sorting.

For a list like 65318724, If we do Merge Sort, the process will be like this:

65 31 87 24　　If the integer moves from right to left, it means that this integer is disorder

6 5 3 1 8 7 2 4 0 count

⬇

65 31 87 42 1 count: ***24->42 (4>2)***

⬇

6531 8742 0 count

⬇

87654321 4 + 4 + 2 + 1 counts

8>6531,7>6531,4>31,2>1

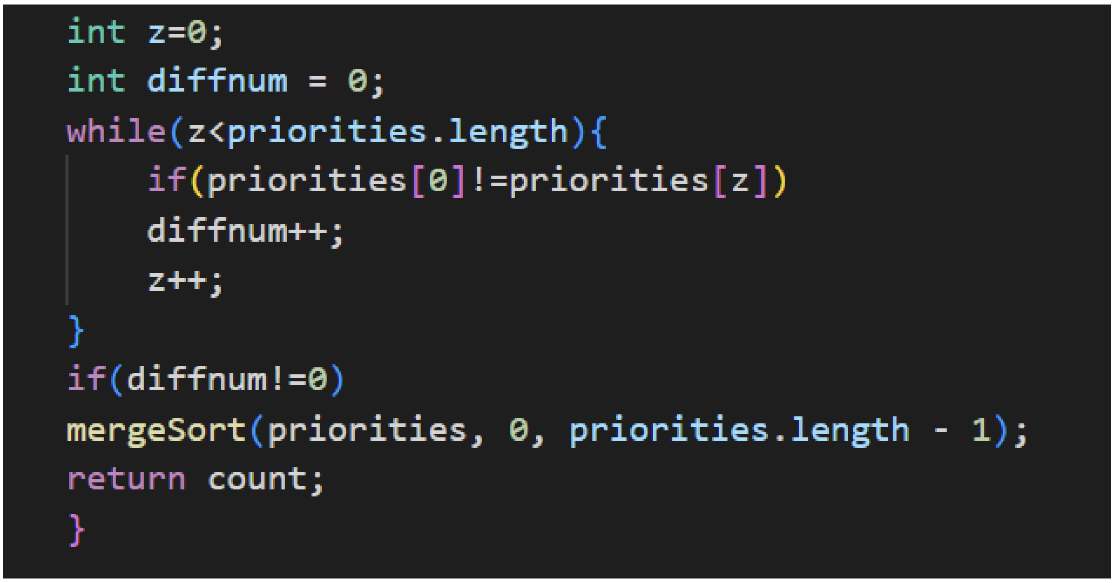
So in total there are 1+4+4+2+1=12 fines in this case.

n1 is the length of the left hand side array, and i is the current position of the integer in left hand side array which is being compared to.

The code line: ***count=count+n1-i***; means in Merge Sort, when integer a from the right hand side array is bigger than integer b in the left hand side array, a will bigger than all the interger in the left hand side array following b (includes b).

The code line: ***count=count+n1-i-1***; means almost the same but b is not included since in this case a=b.

Also, we use the code below to detect whether the list contains the same integer.



```
int z=0;
int diffnum = 0;
while(z<priorities.length){
    if(priorities[0]!=priorities[z])
    diffnum++;
    z++;
}
if(diffnum!=0)
mergeSort(priorities, 0, priorities.length - 1);
return count;
}
```

In conclusion, by implementing Merge Sort, we count the numbers of disorder integer as the sorting process proceeds.

Time Complexity

O(NlogN))