

NAME :: IBRAR BABAR

ROLL NO. :: 19P-0104

SECTION(4A)

ANALYSIS OF ALGORITHM

ASSIGNMENT(3)

	Worstcase	Best case
MergeArray(int arr1[], int arr2[], int n1, int n2, int arr3[]){	1	1
int i=0, j=0, k=0;	1	1
while(i < n1 && j < n2){	$(n/2)+1$	$(n/2)+1$
if(arr1[i] < arr2[j]){	$n/2$	$n/2$
arr3[k] = arr1[i];	$n/2$	$n/2$
k++;	$n/2$	$n/2$
i++;}	$n/2$	$n/2$
else{		
arr3[k] = arr2[j];	$n/2$	$n/2$
k++;	$n/2$	$n/2$
j++;}	$n/2$	$n/2$
while(i < n1)	$(n/2)+1$	0
arr3[k++] = arr1[i++];	$(n/2)$	0
while(j < n2)	$(n/2)+1$	$n/2$
arr3[k++] = arr2[j++];	$n/2$	$n/2$

```
int main() {
```

```
int arr1[] = {80, 85, 86, 87, 100};
```

```
int n1 = sizeof(arr1)/sizeof(arr1[0]);
```

```
int arr2[] = {2, 60, 69, 70, 86, 90};
```

```
int n2 = sizeof(arr2)/sizeof(arr2[0]);
```

```
int n = n1 + n2;
```

```
int arr3[n];
```

```
mergeArray(arr1, arr2, n1, n2, arr3);
```

```
int a = n/2;
```

```
if (a % 2 != 0) {
```

```
cout << arr3[n-1] << " " << arr3[a] << endl; }
```

```
if (a % 2 == 0) {
```

```
cout << arr3[a] cout << arr3[a] << endl; }
```

Worse case

best case

1

1

1

1

1

1

1

1

By ignoring constant and lower order terms

worse case = $\Omega(n)$

Best case = $O(n)$

So in other words.

the running time of this algorithm is

$$\left[\underline{\underline{O(n)}} \right]$$
