Merge Sort

**package** com.hcl.java;

**class** MergeSort {

**void** merge(**int** arr[], **int** l, **int** m, **int** r) {

**int** n1 = m - l + 1;

**int** n2 = r - m;

/\* Create temp arrays \*/

**int** L[] = **new** **int**[n1];

**int** R[] = **new** **int**[n2];

/\* Copy data to temp arrays \*/

**for** (**int** i = 0; i < n1; ++i)

L[i] = arr[l + i];

**for** (**int** j = 0; j < n2; ++j)

R[j] = arr[m + 1 + j];

**int** i = 0, j = 0;

**int** k = l;

**while** (i < n1 && j < n2) {

**if** (L[i] <= R[j]) {

arr[k] = L[i];

i++;

} **else** {

arr[k] = R[j];

j++;

}

k++;

}

**while** (i < n1) {

arr[k] = L[i];

i++;

k++;

}

**while** (j < n2) {

arr[k] = R[j];

j++;

k++;

}

}

**void** sort(**int** arr[], **int** l, **int** r) {

**if** (l < r) {

**int** m = (l + r) / 2;

sort(arr, l, m);

sort(arr, m + 1, r);

merge(arr, l, m, r);

}

}

**static** **void** printArray(**int** arr[]) {

**int** n = arr.length;

**for** (**int** i = 0; i < n; ++i)

System.***out***.print(arr[i] + " ");

System.***out***.println();

}

// Driver method

**public** **static** **void** main(String args[]) {

**int** arr[] = { 12, 11, 13, 5, 6, 7 };

System.***out***.println("Given Array");

*printArray*(arr);

MergeSort ob = **new** MergeSort();

ob.sort(arr, 0, arr.length - 1);

System.***out***.println("\nSorted array");

*printArray*(arr);

}

}

What is big O for merge sort algorithm?