|  |
| --- |
| 0hnQGAuNPo4wojnr77c2x8cWiaT70FGHBZ2h-3__FYQ.png |
| Merge Sort |
| Data Structures Made Easy |
|  |

Dublin city university

# 1. *Merge Sort*

class merge\_Sort{

public static void sort(int [] input){

mergeSort(input, 0, input.length - 1);

}

public static void mergeSort(int [] array, int low, int high){

merge\_Sort\_Test\_Program test = new merge\_Sort\_Test\_Program();

if(low < high){

int mid = (low + high)/2;

mergeSort(array, low, mid);

mergeSort(array, mid + 1, high);

merge(array, low, mid, high);

test.printArray(array);

}

}

public static void merge(int [] array, int low, int mid, int high){

int [] temporary = new int[high - low + 1];

int left = low;

int right = mid + 1;

int element = 0;

while(left <= mid && right <= high){

if(array[left] < array[right]){

temporary[element] = array[left];

left++;

}

else{

temporary[element] = array[right];

right++;

}

element++;

}

if(left <= mid){

while(left <= mid){

temporary[element] = array[left];

left++;

element++;

}

}

else if(right <= high){

while(right <= high){

temporary[element] = array[right];

right++;

element++;

}

}

for(int index = 0; index < temporary.length; index++)

array[index + low] = temporary[index];

}

}

# 2. *Merge Sort Test Program*

class merge\_Sort\_Test\_Program{

public static int [] readInputArray(){

int [] merge\_Array = {80, 32, 56, 37, 69, 76};

return merge\_Array;

}

public static void printArray(int [] array){

for(int index = 0; index < array.length; index++)

System.out.print(array[index] + " , ");

System.out.println();

}

public static void main(String [] args){

merge\_Sort merge = new merge\_Sort();

int [] input\_Array = readInputArray();

System.out.println('\n' + "INPUT ARRAY");

System.out.println("===========");

printArray(input\_Array);

merge.sort(input\_Array);

System.out.println('\n' + "SORTED ARRAY");

System.out.println("============");

printArray(input\_Array);

}