Merge sort algorithm

|  |
| --- |
|  |
|  | #include <iostream>  using namespace std; |
|  | void swapping(int &x, int &y) { //swapping the content of x and y |
|  | int temp; |
|  | temp = x; |
|  | X = y; |
|  | y = temp; |
|  | } |
|  | void display(int \*array, int size) { |
|  | for(int i = 0; i<size; i++) |
|  | cout << array[i] << " "; |
|  | cout << endl; |
|  | } |
|  | void merge(int \*array, int l, int m, int r) { |
|  | int i, j, k, nl, nr; |
|  | //size of left and right sub-arrays |
|  | nl = m-l+1; nr = r-m; |
|  | int larr[nl], rarr[nr]; |
|  | //fill left and right sub-arrays |
|  | for(i = 0; i<nl; i++) |
|  | larr[i] = array[l+i]; |
|  | for(j = 0; j<nr; j++) |
|  | rarr[j] = array[m+1+j]; |
|  | i = 0; j = 0; k = l; |
|  | //marge temp arrays to real array |
|  | while(i < nl && j<nr) { |
|  | if(larr[i] <= rarr[j]) { |
|  | array[k] = larr[i]; |
|  | i++; |
|  | }else{ |
|  | array[k] = rarr[j]; |
|  | j++; |
|  | } |
|  | k++; |
|  | } |
|  | while(i<nl) { //additional element in left array |
|  | array[k] = larr[i]; |
|  | i++; k++; |
|  | } |
|  | while(j<nr) { //additional element in right array |
|  | array[k] = rarr[j]; |
|  | j++; k++; |
|  | } |
|  | } |
|  | void mergeSort(int \*array, int l, int r) { |
|  | int m; |
|  | if(l < r) { |
|  | int m = l+(r-l)/2; |
|  | // Sort first and second arrays |
|  | mergeSort(array, l, m); |
|  | mergeSort(array, m+1, r); |
|  | merge(array, l, m, r); |
|  | } |
|  | } |
|  | int main() { |
|  | int n; |
|  | cout << "Enter the number of elements: "; |
|  | cin >> n; |
|  | int arr[n]; //create an array with given number of elements |
|  | cout << "Enter elements:" << endl; |
|  | for(int i = 0; i<n; i++) { |
|  | cin >> arr[i]; |
|  | } |
|  | cout << "Array before Sorting: "; |
|  | display(arr, n); |
|  | mergeSort(arr, 0, n-1); //(n-1) for last index |
|  | cout << "Array after Sorting: "; |
|  | display(arr, n); |
|  | } |