|  |
| --- |
| public static void mergeSort(int[] a ) {         int[] tmpArray = new int[a.length];          mergeSort( a, tmpArray, 0, a.length - 1 );    }            private static void mergeSort( int[] a, int[] tmpArray, int left, int right ) {         if( left < right ) {             int center = ( left + right ) / 2;             mergeSort( a, tmpArray, left, center );             mergeSort( a, tmpArray, center + 1, right );             merge( a, tmpArray, left, center + 1, right );         }    }            private static void merge( int[] a, int[] tmpArray, leftPos, int rightPos, int rightEnd ) {         int leftEnd = rightPos - 1;         int tmpPos = leftPos;         int numElements = rightEnd - leftPos + 1;                         while( leftPos <= leftEnd && rightPos <= rightEnd )             if( a[ leftPos ]  <  a[ rightPos ] ){                 tmpArray[ tmpPos] = a[ leftPos]; tempos++; leftPos++;\ }              else{ tmpArray[ tmpPos ] = a[ rightPos ]; tmpPos++; rightPos++; } }                 while( leftPos <= leftEnd ) {                 tmpArray[ tmpPos ] = a[ leftPos ]; tmpPos++; leftPos++; }                  while( rightPos <= rightEnd ) {               tmpArray[ tmpPos ] = a[ rightPos ]; tmpPos++; rightPos++; }                 for( int i = 0; i < numElements; i++, rightEnd-- )             a[ rightEnd ] = tmpArray[ rightEnd ];    }  |
|  |