

Problem 11858 "Frosh week" explanation

viernes, 9 de abril de 2021 11:39

Code:

TotalSwaps = 0;

MergeSort(int l, int r, Vector v){

If(l < r){

Mid = (l + r)/2;

Mergesort(v, lb, mid);

Mergesort(v, mid+1, ub);

Merge(v, l, mid, r);

}

}

Merge(Vector v, int left , int right, int mid){

// ordena dos arreglos ordenados

Vector A,B;

n1: m - l + 1;

n2: r - m;

For(i = 0 to n1)

A.push <- v[l + i]

For(j = 0 to n2)

B.push <- v[m + 1 + j]

K = l;

While (i < n1 && j < n2){

If (A[i] <= B[j])

v[k] = A[i];

i++;

Else

v[k] = B[j]

j++

k++

}

While (i < n1)

v[k] = A[i]

i++, k++

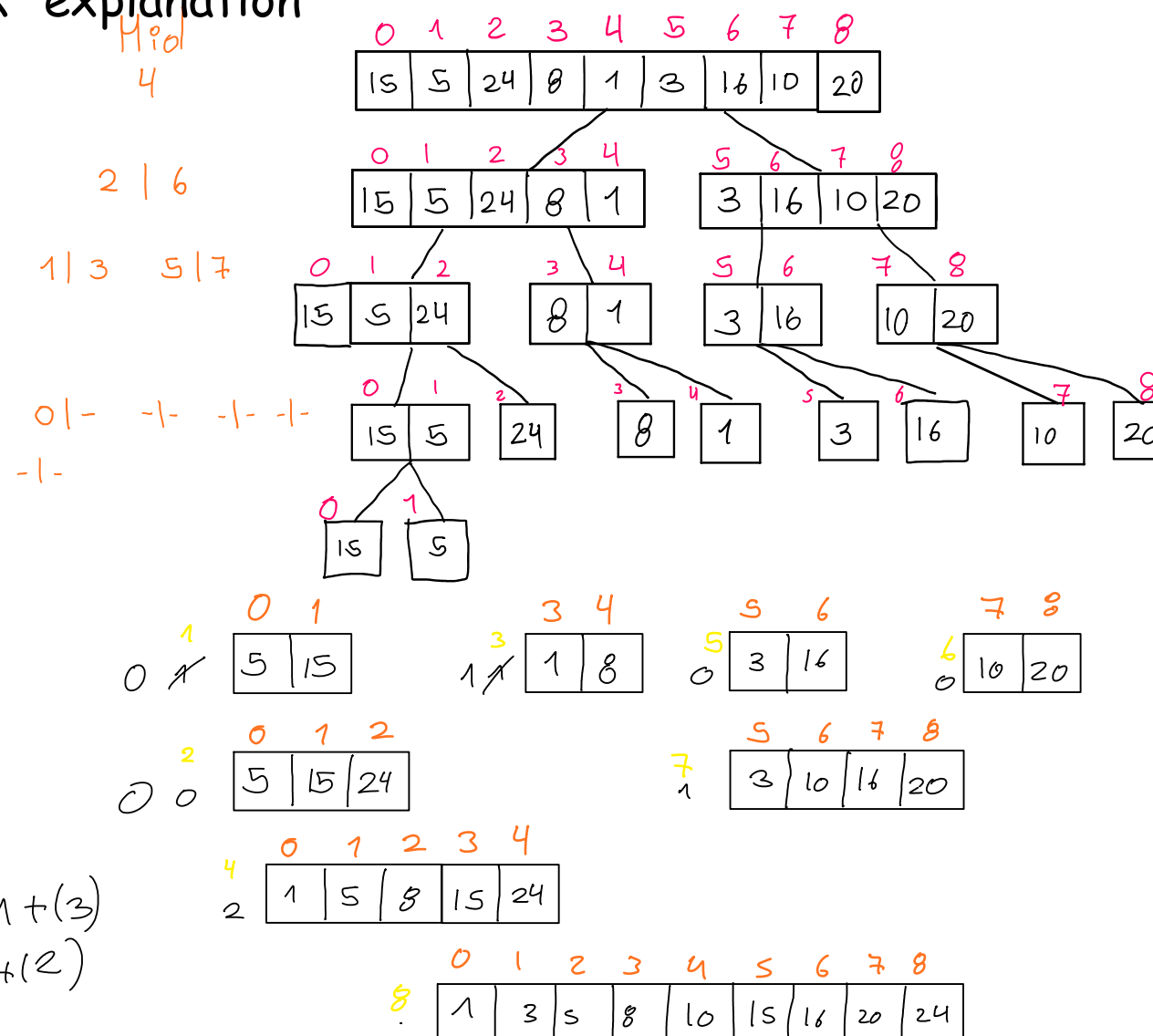
While (j < n2)

v[k] = B[j]

j++, k++

}

Answer: print (totalSwaps);



Imaginemos que el vector de A sólo tiene sólo un elemento mayor (Llamemoslo A.maxvar) que algún elemento de B (llamemoslo B.minvar).

Por la naturaleza de merge sort sabemos que el vector de la izquierda se encuentra ordenado de manera ascendente.

Por lo tanto, al momento de realizar un swap entre A.maxvar y B.minvar, debemos ser cuidadosos en no desordenar ambos arreglos, por lo que debemos realizar swaps de manera lineal entre B.minvar y algún elemento de menor índice en arreglo hasta llegar a la posición A.maxvar, lo que se traduciría en:

Numswaps = A.length - A.maxvar.pos;

Totalswaps += Numswaps;

- No se considera el largo de B puesto que se asume que los elementos de B que sean de mayor tamaño que los de A ya han sido ubicados en el arreglo y para los restantes basta con agregarlos de manera lineal.