

03/07/20

## Algorithm :-

Step 1: start

Step 2: Input  $a[] = \{64, 82, 45, 125, 15, 26, 1, 2, 25, 25\}$

Step 3:  $\text{rmerge}(\text{Int} **a, \text{int } i\text{begin}, \text{int } i\text{mid}, \text{int } i\text{end}, \text{int} **b)$

Step 4:  $\text{splitmerge}(\text{Int} **b, \text{int } i\text{begin}, \text{int } i\text{end}, \text{int} **a)$

Step 5:  $**\text{merge sort}(\text{Int} **a, \text{int } \text{size})$

Step 6:  $\text{Int } \text{size} = \text{size of } a / \text{size of } a[0]$

Step 7:  $\text{Int} **\text{ret} = \text{merge sort}(a, \text{size})$

Step 8: Display sorted array element.

Step 9: for  $(\text{int } i = 0; i < \text{size}; i++)$   
Display  $*\text{ret}[i]$

Step 10: free(ret)

Step 11: stop.

$\text{merge}(\text{Int} **a, \text{int } i\text{begin}, \text{int } i\text{mid}, \text{int } i\text{end}, \text{Int} **b)$

Step 1: Entry

Step 2:  $\text{int } i = i\text{begin}, j = i\text{mid}$

Step 3: for  $(\text{int } k = i\text{begin}; k < i\text{end}; k++)$

$b[k] = a(i < \text{mid} \ \&\& \ (j = i\text{end} \ || \ *a[i] < *a[j]) i++ ; j++)$

Step 4: End.

$\text{split merge}(\text{Int} **b, \text{int } i\text{begin}, \text{int } i\text{end}, \text{Int} **a)$

Step 1: Entry

Step 2: if  $(i\text{end} - i\text{begin} < 2)$   
return

Step 3:  $\text{int } i\text{mid} = (i\text{end} + i\text{begin}) / 2$

Step 4:  $\text{split merge}(a, i\text{begin}, i\text{mid}, b)$

Step 5:  $\text{split merge}(a, i\text{mid}, i\text{end}, b)$

Step 6:  $\text{merge}(b, i\text{begin}, i\text{mid}, i\text{end}, a)$

Step 7: End.

int merge sort (int \*a, int size)

STEP 1: Entry.

STEP 2: Int \*\*ret = malloc (size \* size of \*ret)

STEP 3: Int \*tmp = malloc (size \* size of \*tmp)

STEP 4: for (int i=0; i < size; i++)  
ret[i] = tmp[i] = a[i]

STEP 5: split merge (tmp, 0, size, ret)

STEP 6: free (tmp)

STEP 7: return ret.

Flowchart

Start



Input a[] = {64, 32, 45, 125, 15, 265, 1, 2, 25, 25}



merge (int \*\*a, int begin, int mid, int end, int \*\*b)



split merge (int \*\*b, int begin, int end, int \*\*a)



\*\* merge sort (int \*a, int size)



int size = size of a / size of a[0]



Int \*\*ret = merge sort (a, size)

Display sorted array element



for (int i=0; i < size; i++)



Display \*ret[i]



free (ret)



Stop

True

merge (int \*\*a, int ibegin, int imid, int iend, int \*\*b)

Entry

int i = ibegin, j = imid

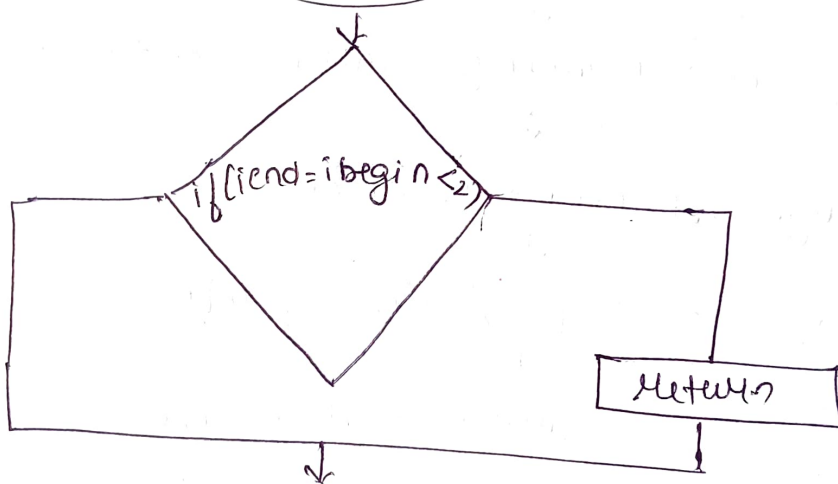
for (int k = ibegin; k < iend; k++)

b[k] = a[i < imid & (j >= iend || \*a[i] <= \*a[j]) ? i++ : j++]

End

\* split merge (int \*\*b, int ibegin, int iend, int \*\*a)

Entry



int imid = (iend + ibegin) / 2

split merge (a, ibegin, imid, b)

split merge (a, imid, iend, b)

merge (b, ibegin, imid, iend, a)

End

**\*\* merge sort (int \*a, int size)**

Entry

Int \*\* ret = malloc (size \* size of \*ret)

Int \*\* tmp = malloc (size \* size of \*tmp)

for (int i = 0; i < size; i++)

ret[i] = tmp[i] = a + i

split merge (tmp, 0, size, ret)

free (tmp)

return ret