```
num[] = {8, 5, 3}

mergesort top: low=0, high=2
    0 < 2  // True
        mid = 0+2/2 = 1
        mergesort left subarray: low=0, high=1
            0 < 1  // True
            mid = 0+1/2 = 0  (integer division)
            mergesort left subarray: low=0, high=0
               0 < 0  // False (down to single element and not enter if block)
            mergesort right subarray: low=1, high=1
               1 < 1 // False (down to single element and not enter if block)
            merge: l1=0, u1=0, l2=1, u2=1
               i=0, j=1, k=0
               i <= u1 && j <= u2:  0 <= 0 && 1 <= 1  // True have element in both
                      if (a[i] <= a[j]): a[i]=8 a[j]=5
                                8 <= 5 // False so take else part
                                temp[k] = 5
                                j=2
                                k=1
               i <= u1 && j <= u2:  0 <= 0 && 2 <= 1  // False no element in right
               while for left: 0 <= 0
                   temp[1] = a[0] = 8
                   k=2, i=1
                   // loop again: 1 <= 0 False and come out
               while for right: 2 <= 0  // false
               for loop to copy: temp={5, 8}  a={5, 8, 3} so [0, 1] indices sorted
        mergesort right: low=2, high=2
            1 < 1 // False (down to single element and not enter if block)
        merge: l1=0, u1=1, l2=2, u2=2
            i=0, j=2, k=0
            i <= u1 && j <= u2:  0 <= 1 && 2 <= 2  // True have element in both
                if (a[i] <= a[j]): a[i]=5 a[j]=3
                     5 <= 3  // False so take else part
                     temp[0] = 3
                     j=3
                     k=1
            i <= u1 && j <= u2:  0 <= 1 && 3 <= 2  // False no element in right
            while for left: 0 <= 1
                temp[1] = a[0] = 5,   k=2, i=1
                temp[2] = a[1] = 8,   k=3, i=2
            while for right: 3 <= 2  // false
            for loop to copy:  temp={3, 5, 8}  a={3, 5, 8} so [0, 1, 2] indices sorted
```