

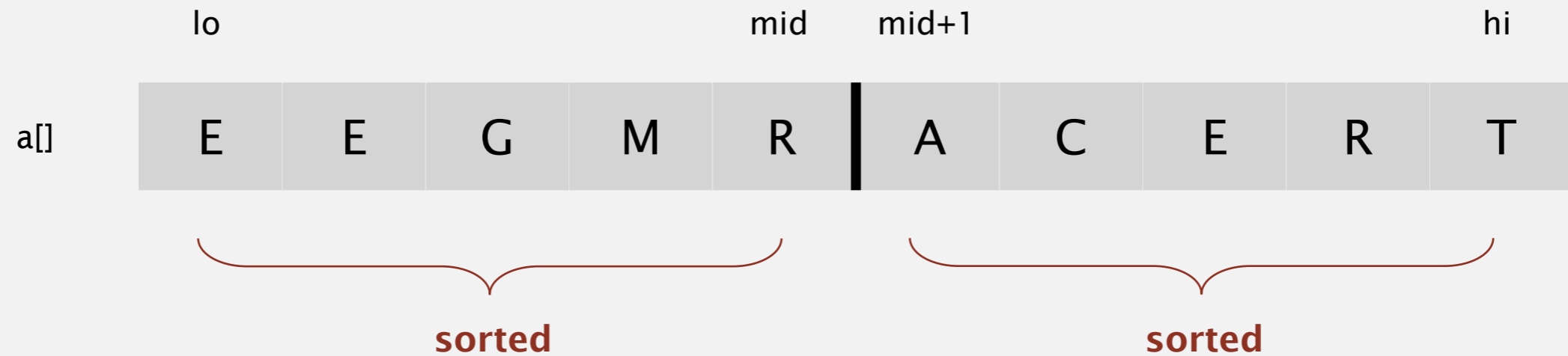
# 2.2 ABSTRACT IN-PLACE MERGE DEMO



[click to begin demo](#)

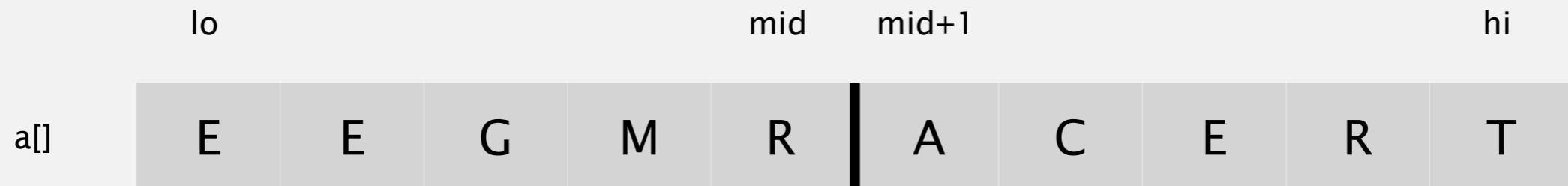
## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[lo]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[lo]$  to  $a[hi]$ .

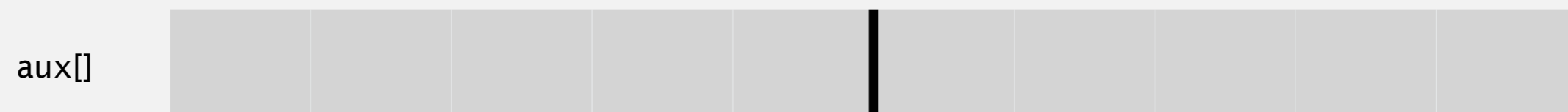


## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[lo]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[lo]$  to  $a[hi]$ .

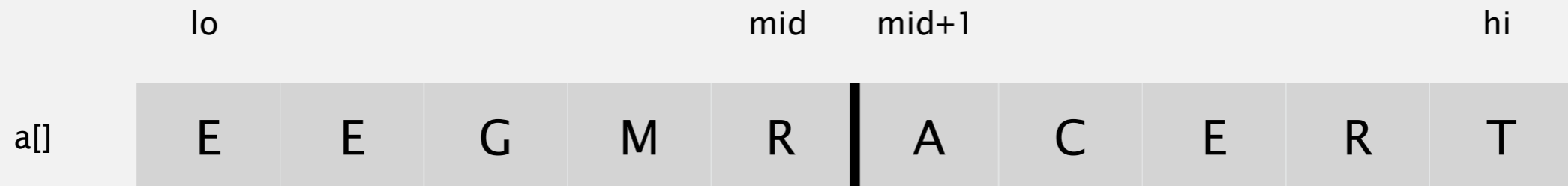


**copy to auxiliary array**



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[lo]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[lo]$  to  $a[hi]$ .



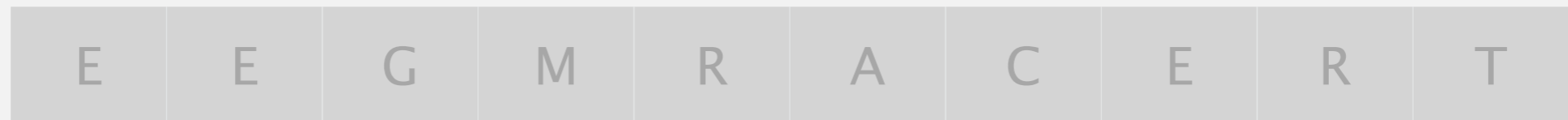
**copy to auxiliary array**



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .

a[]

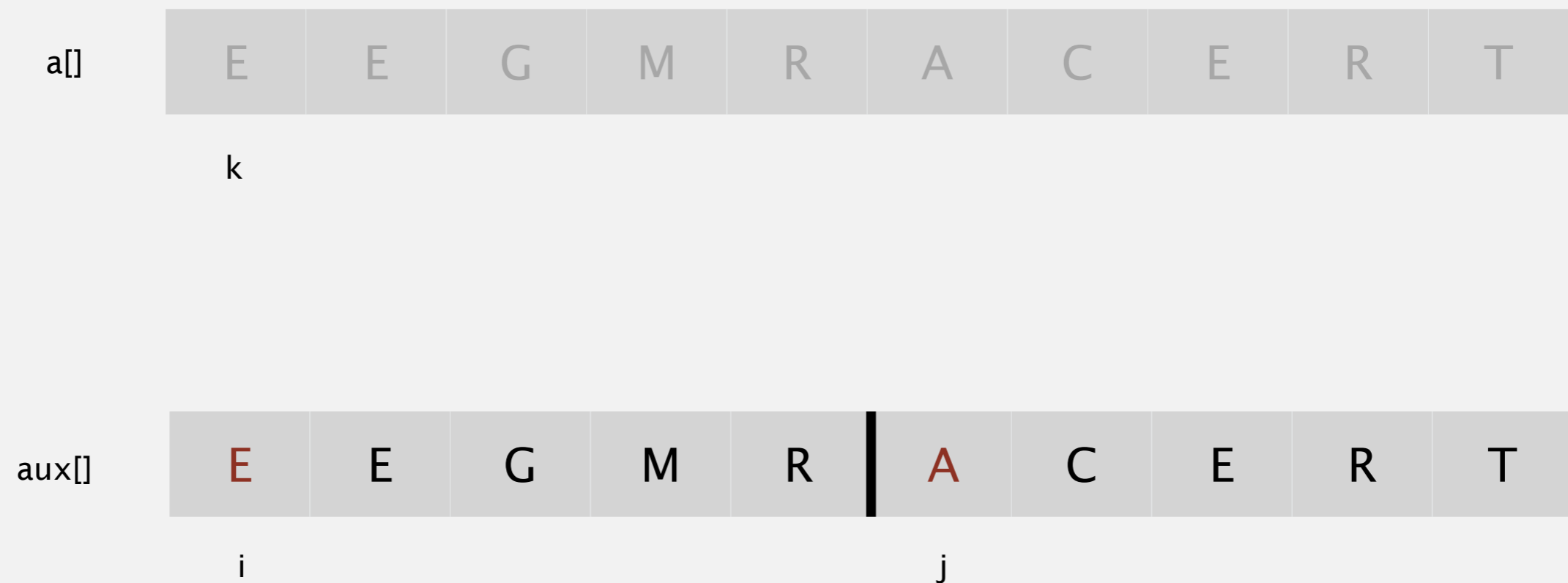


aux[]



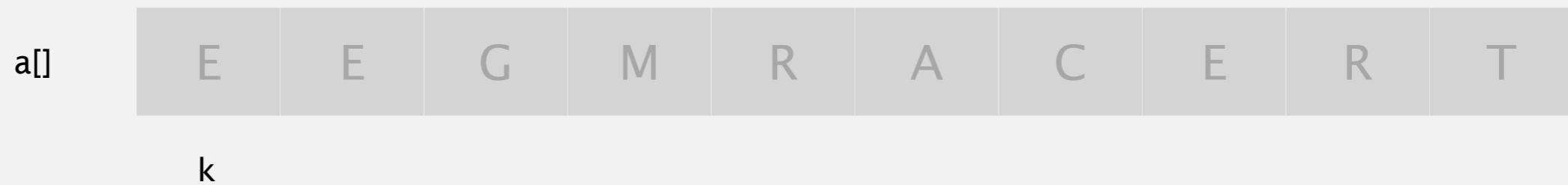
## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .

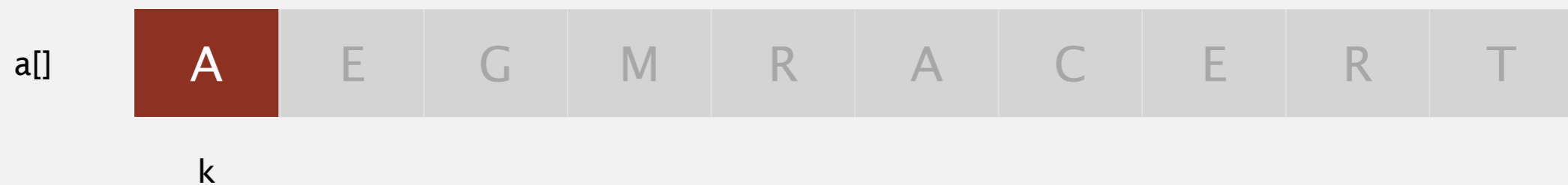


**compare minimum in each subarray**



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



**compare minimum in each subarray**





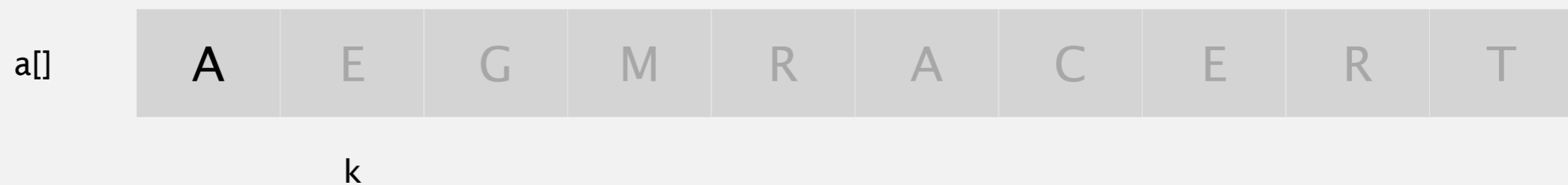
## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .

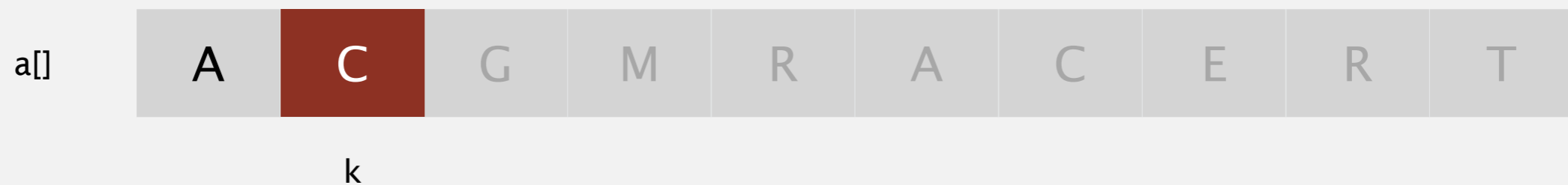


**compare minimum in each subarray**



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



**compare minimum in each subarray**



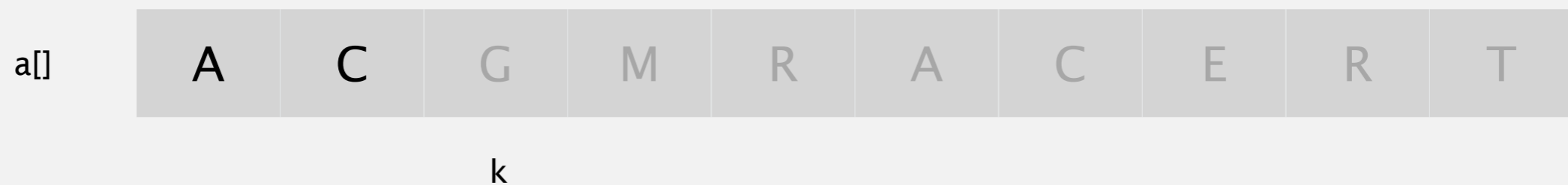
## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .

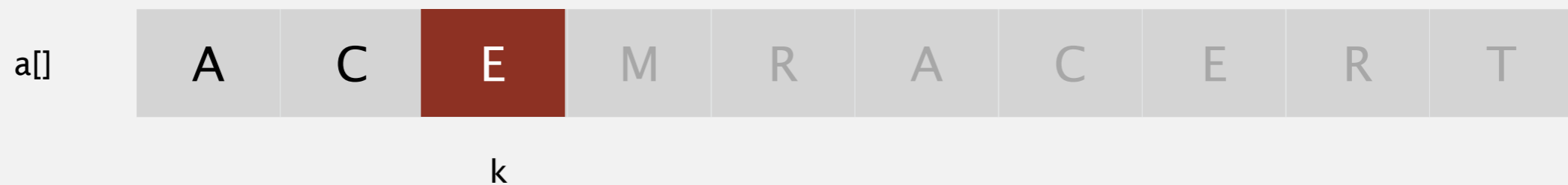


**compare minimum in each subarray**



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



**compare minimum in each subarray**



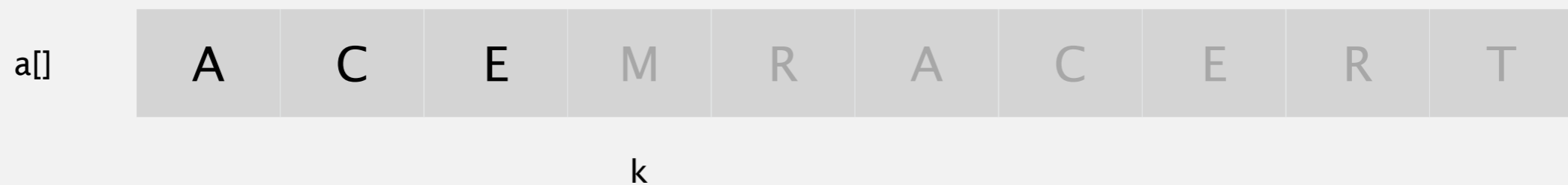
## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



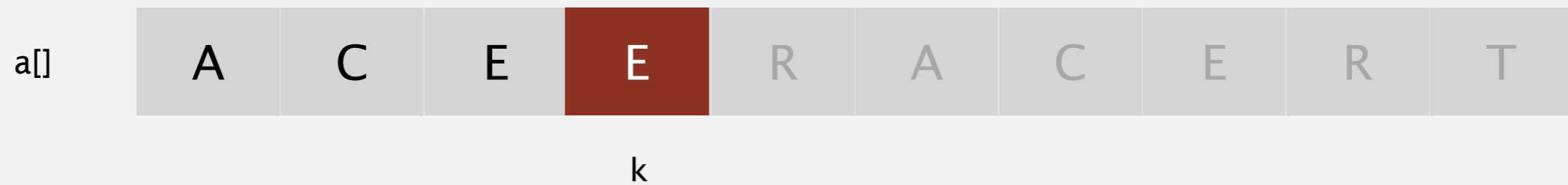
**compare minimum in each subarray**





## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



**compare minimum in each subarray**



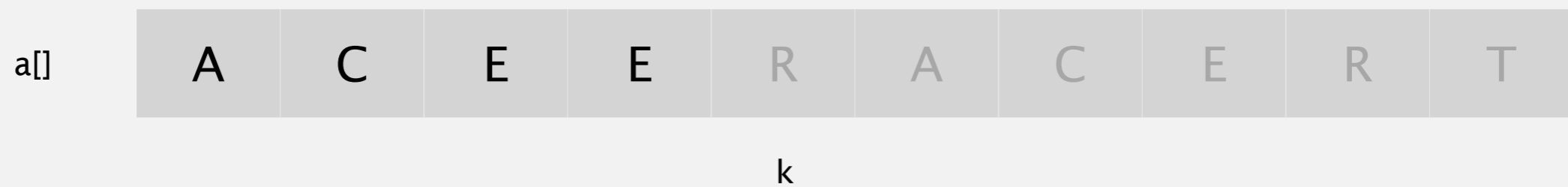
## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[lo]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[lo]$  to  $a[hi]$ .



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .

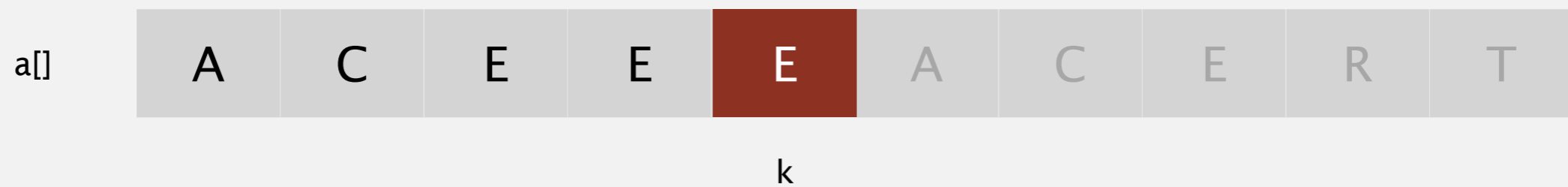


**compare minimum in each subarray**



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[lo]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[lo]$  to  $a[hi]$ .

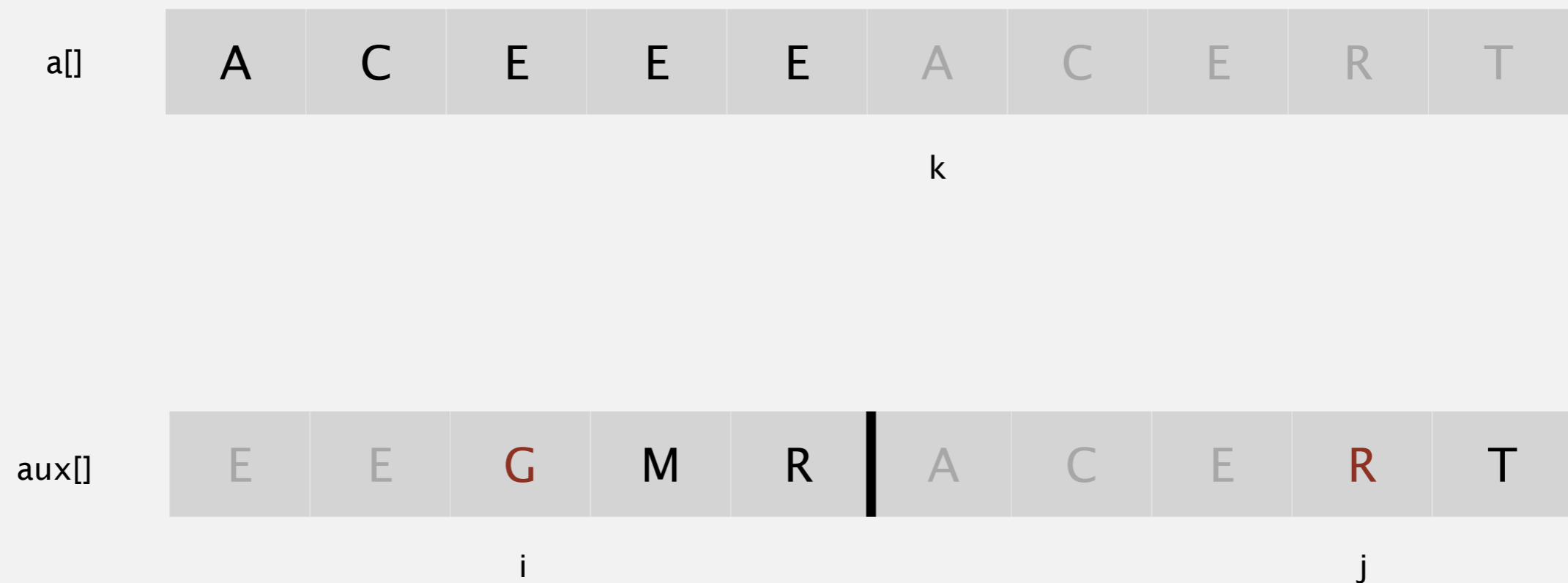


**compare minimum in each subarray**



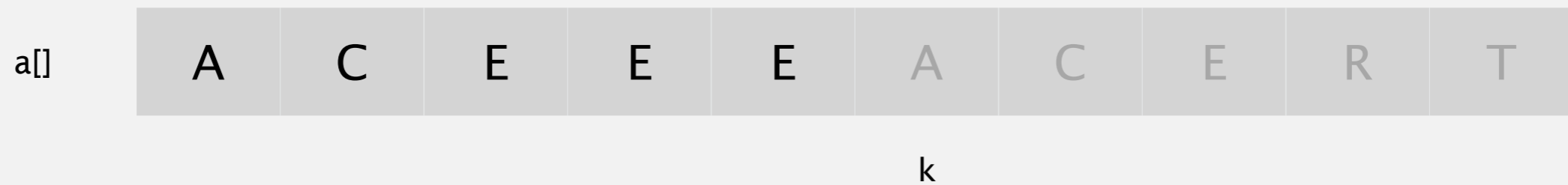
## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



compare minimum in each subarray



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



**compare minimum in each subarray**



## Abstract in-place merge

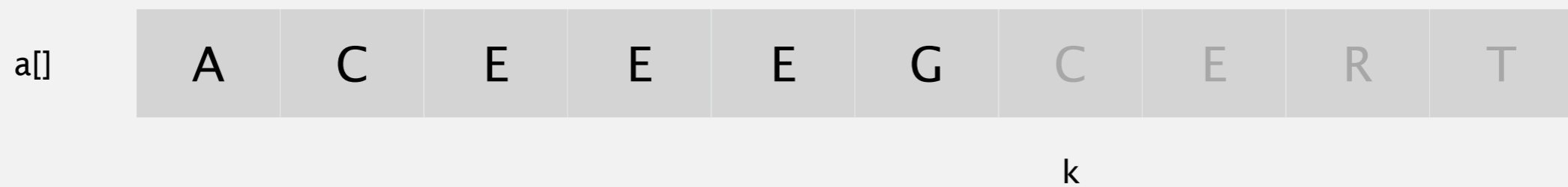
*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .





## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .

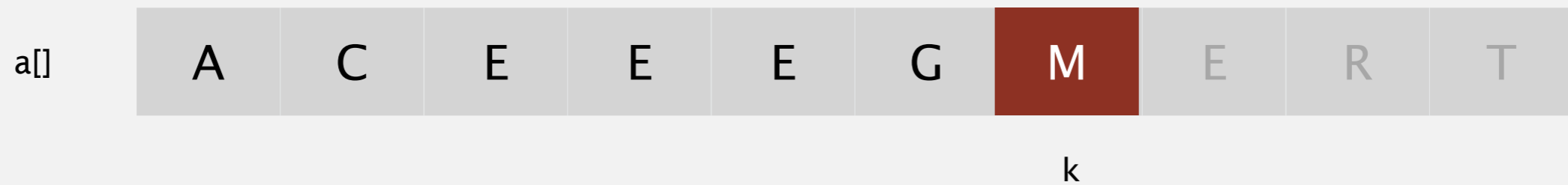


**compare minimum in each subarray**



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .

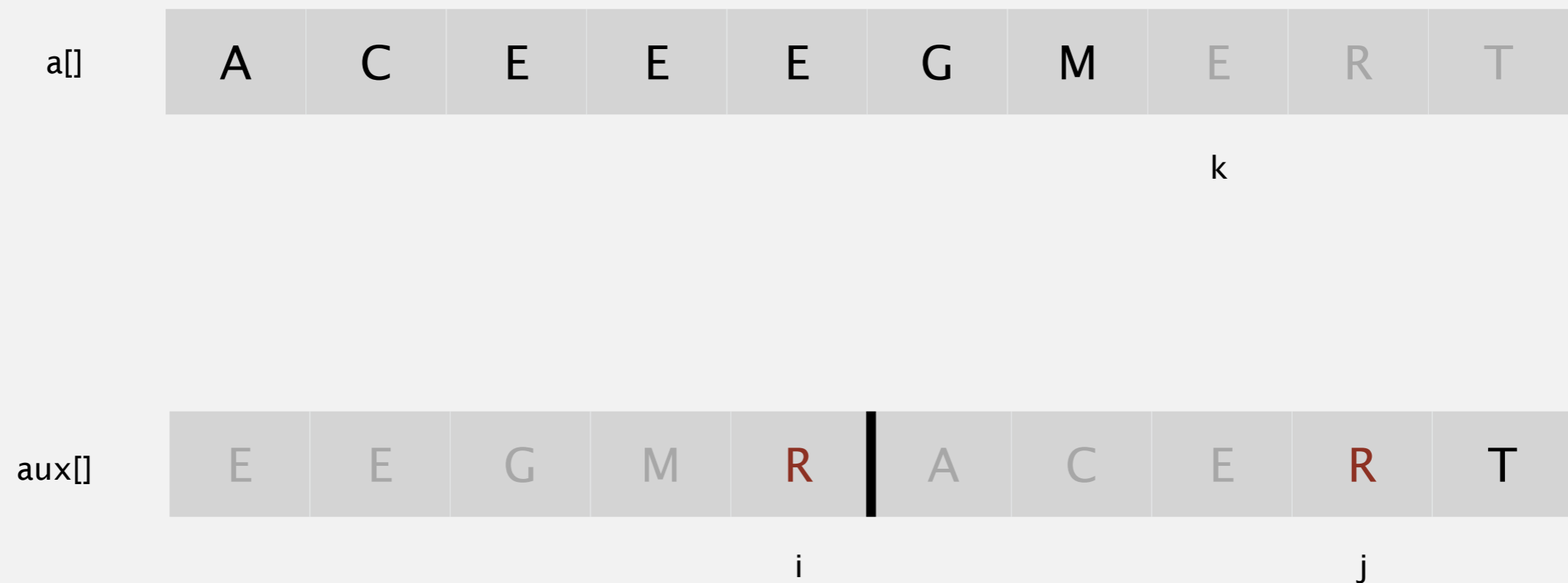


**compare minimum in each subarray**



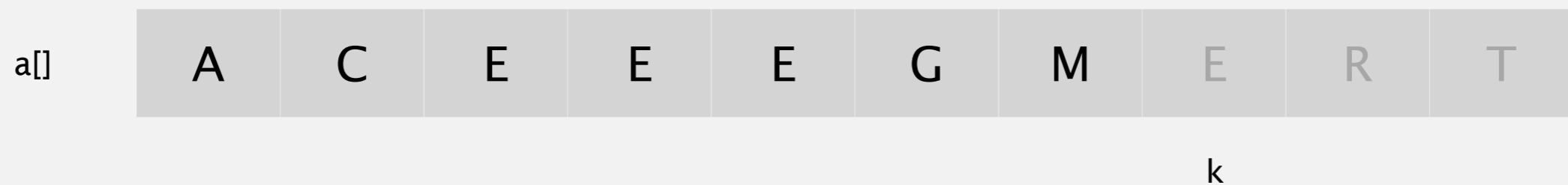
## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .

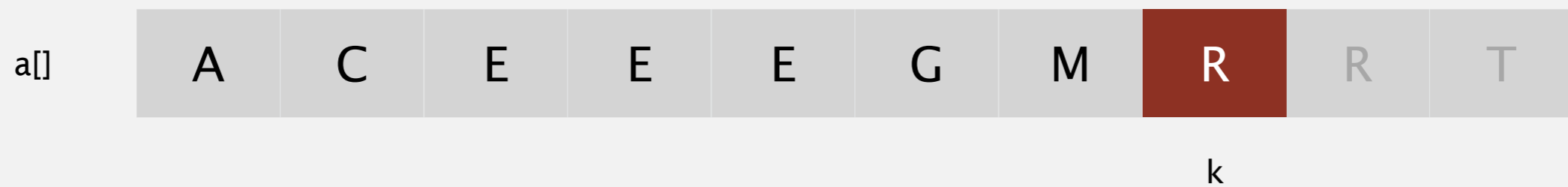


**compare minimum in each subarray**



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



**compare minimum in each subarray**



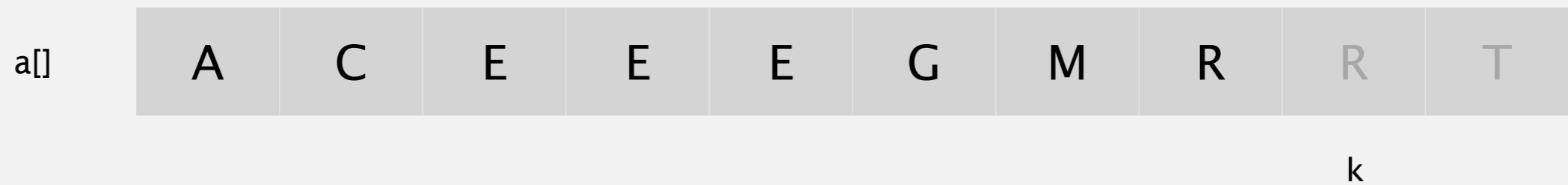
## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[lo]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[lo]$  to  $a[hi]$ .



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .

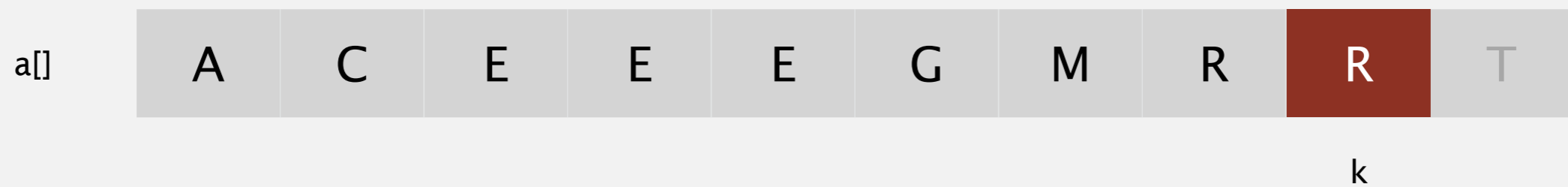


**one subarray exhausted, take from other**



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



**one subarray exhausted, take from other**





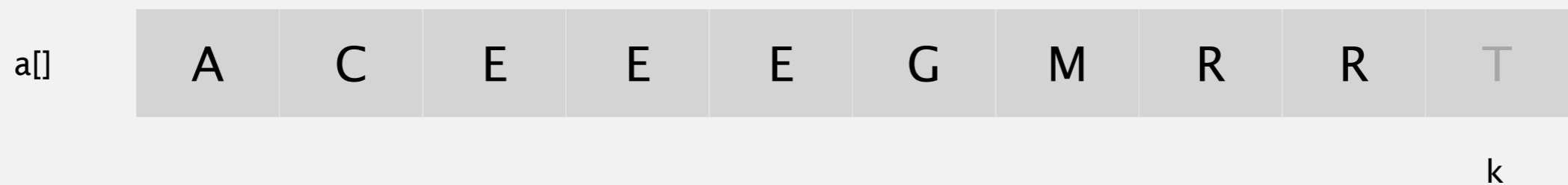
## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .

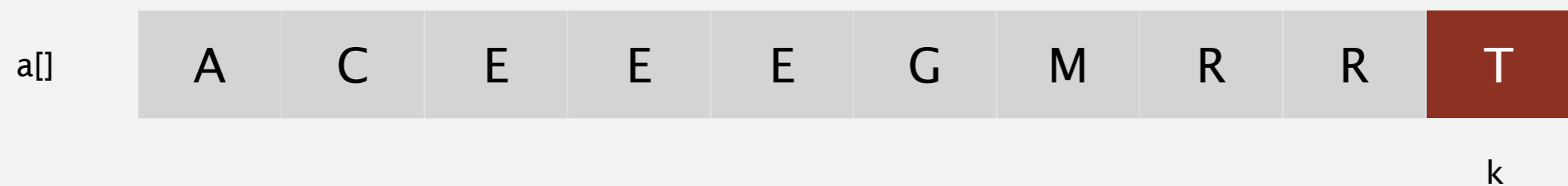


**one subarray exhausted, take from other**



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



**one subarray exhausted, take from other**



## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[l_0]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[l_0]$  to  $a[hi]$ .



k

**both subarrays exhausted, done**



i

j

## Abstract in-place merge

*Goal.* Given two sorted subarrays  $a[lo]$  to  $a[mid]$  and  $a[mid+1]$  to  $a[hi]$ , replace with sorted subarray  $a[lo]$  to  $a[hi]$ .

