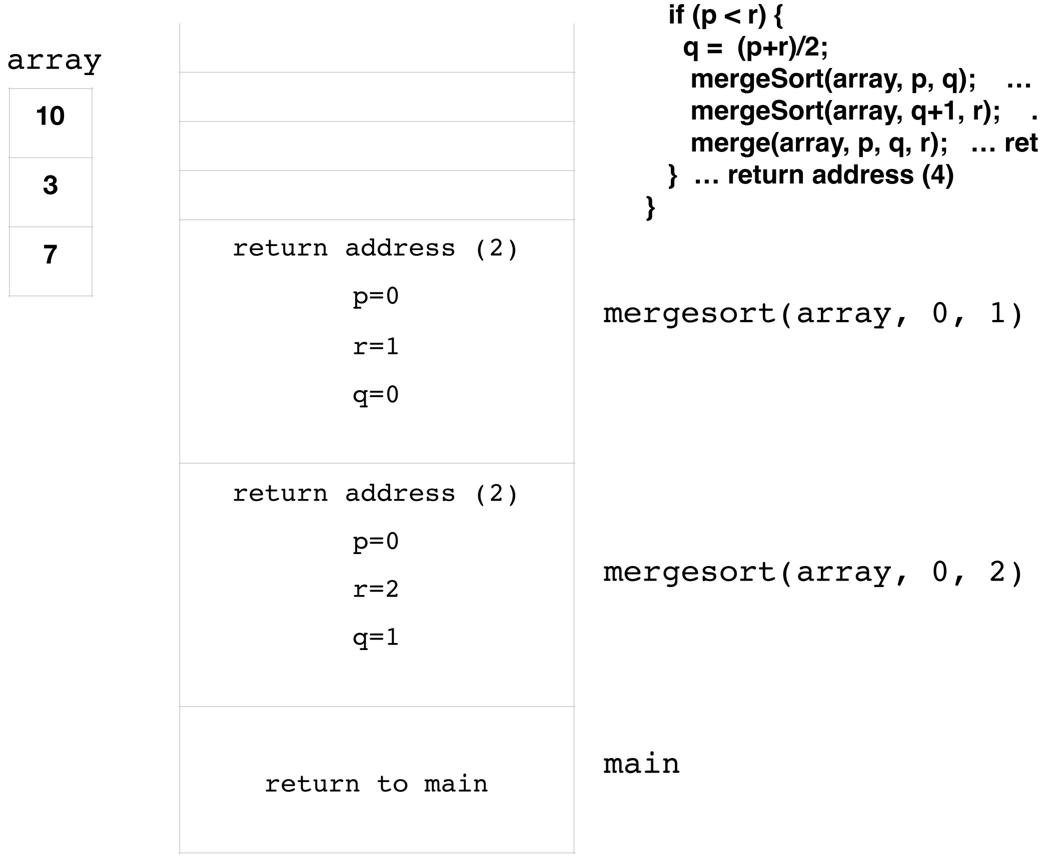
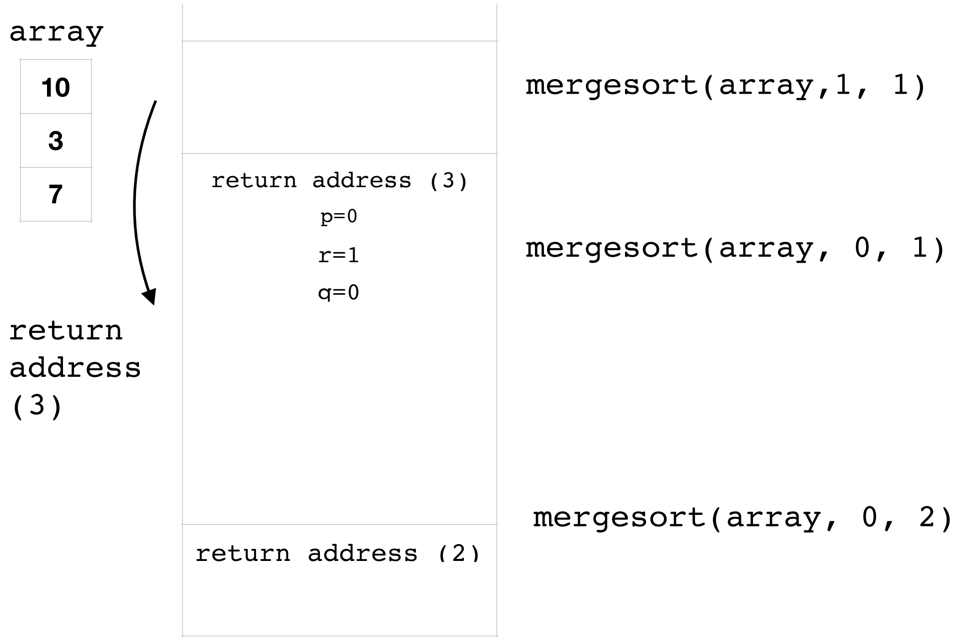
Assignment 1 solution - problems 1

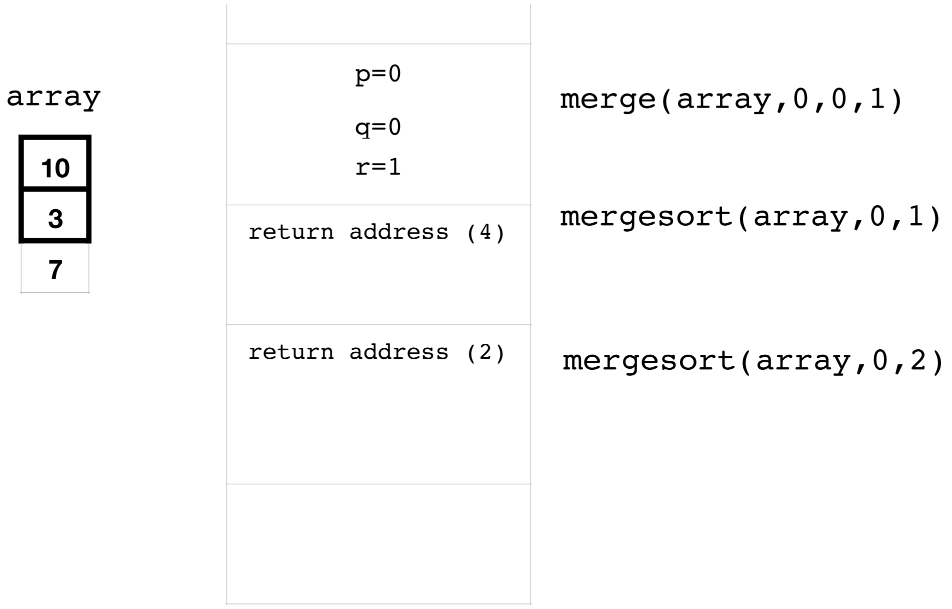
Solution for Problem 1

```
// Merge sort algorithm
mergeSort(array, p, r){
  if (p < r) {
    q =  (p+r)/2;
     mergeSort(array, p, q);    … return address (1)
     mergeSort(array, q+1, r);    … return address (2)
     merge(array, p, q, r);   … return address (3)
  }  … return address (4)
}
```

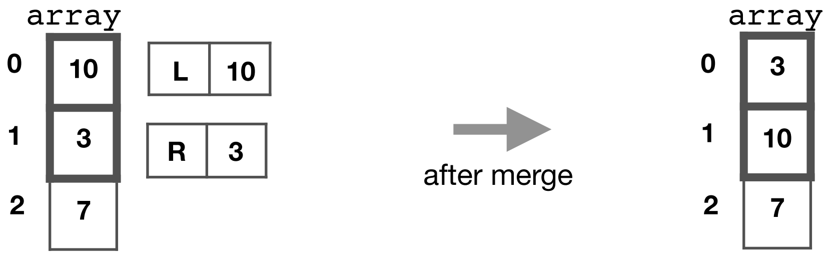


1. `mergesort(array, 0, 1)` calls `mergeSort(array, 0, 0), which` returns because `p < r` is false where `p=0, r=0`
2. `mergeSort(array, 0, 1)` will then call the second `mergeSort(array, q+1,r):`

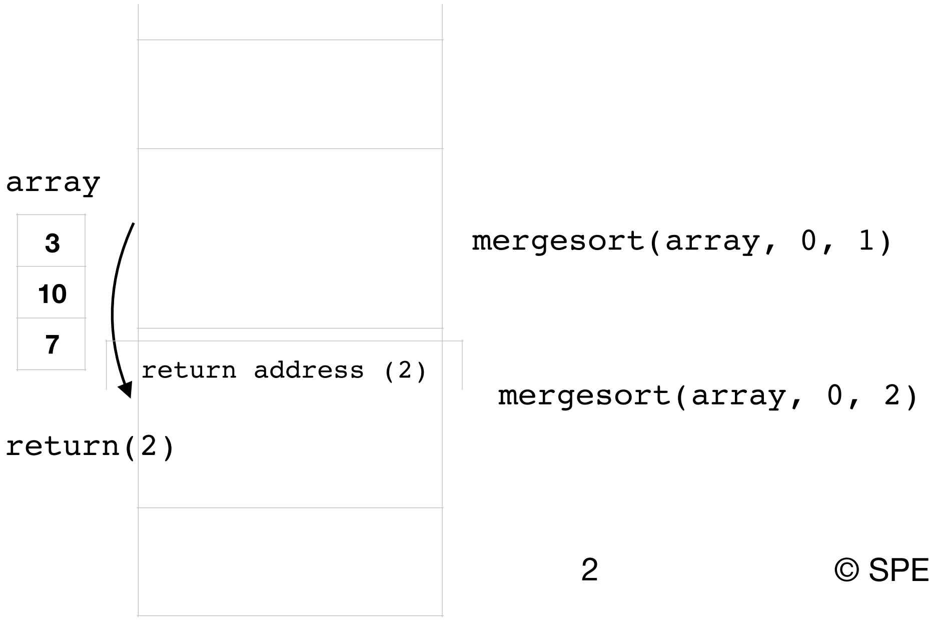




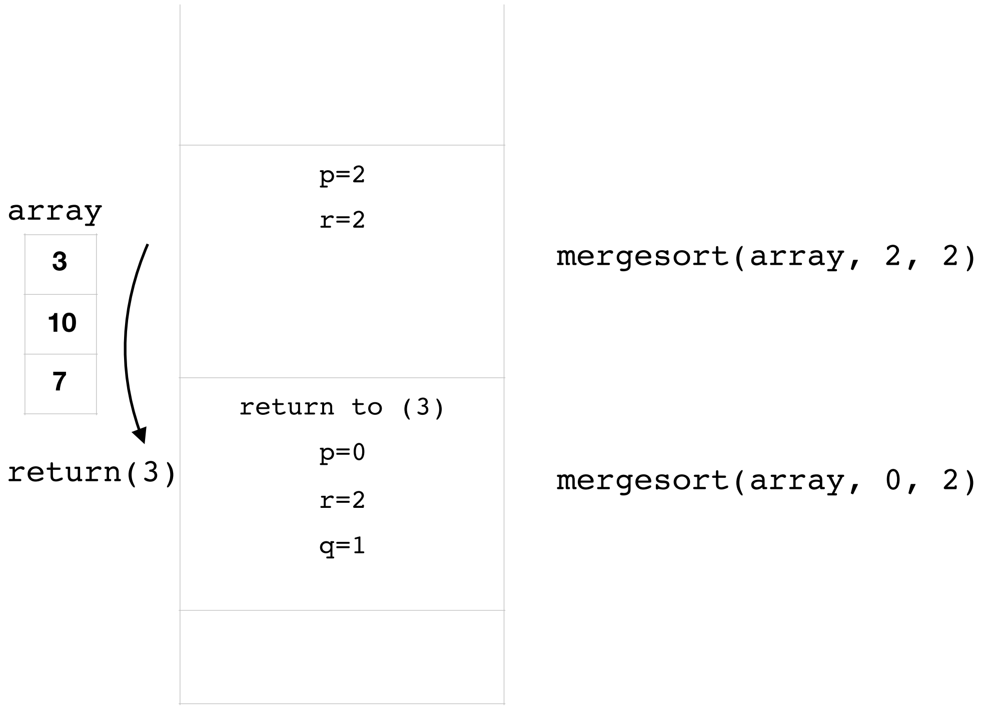
`merge(array, 0, 0, 1)` will copy the data from array to temporary arrays L and R, and then copy it back to array in the correct order.

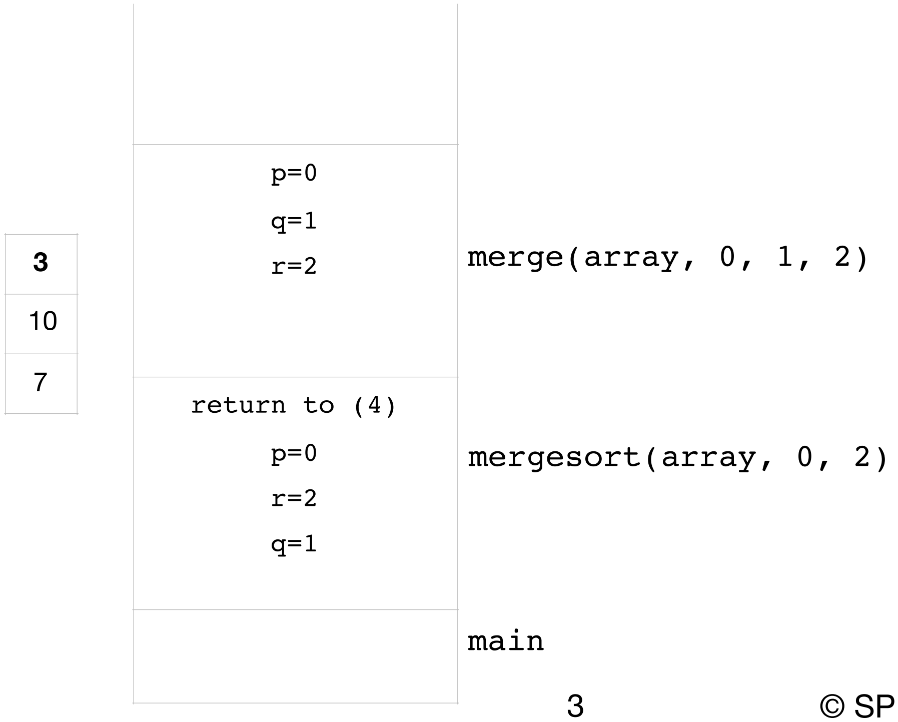


After this `merge(array, 0, 0, 1)` returns to `mergesort(array, 0, 1)`

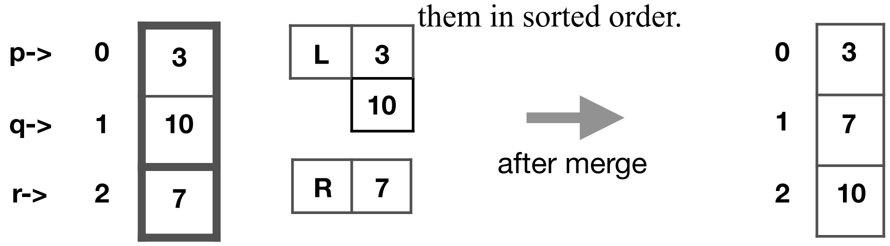


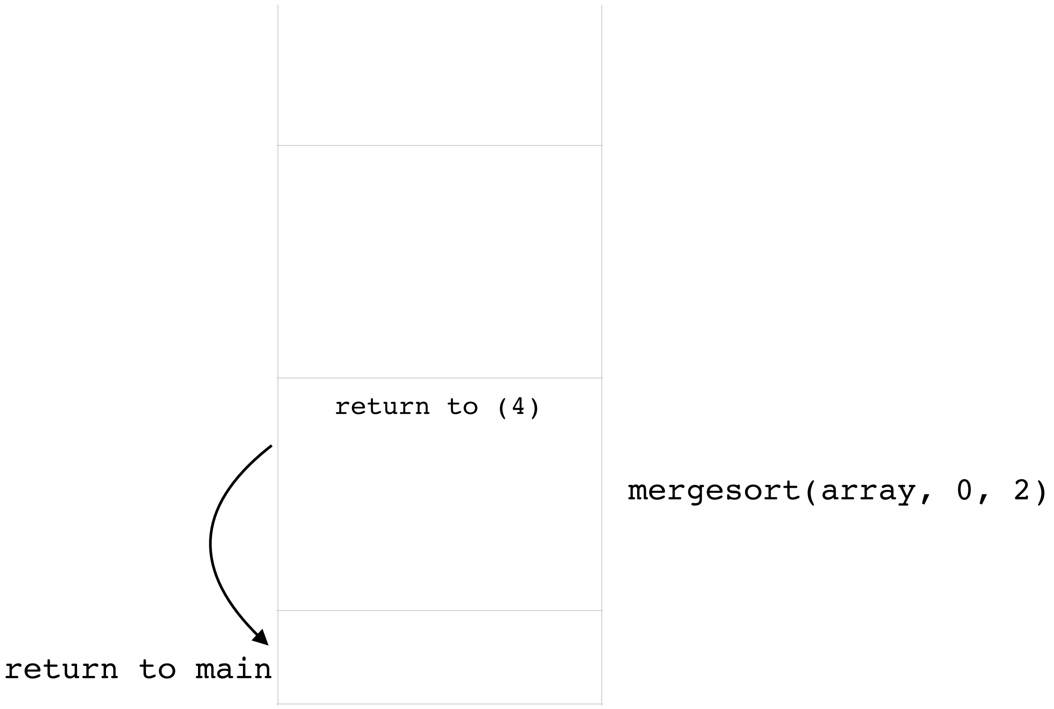
`mergesort(array, 0, 1)` has completed and returned to `address(2)` in `mergesort(array, 0, 2)`, as shown in previous diagram. `mergesort(array, 2, 2)is` executed if `(p < r) -` this is false and so it returns to `(3)` in caller.



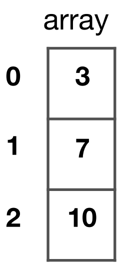


`merge(array, 0, 1, 2)` will execute now. It will create two arrays L and R and merge them in sorted order.





After the merge is completed, it will return to the caller.
`mergesort(array, 0, 2)` has completed and it return to main. The final array contents are :


array
0 3
1 7
2 10