Feedback — Mergesort

Help Center

You submitted this quiz on **Mon 21 Sep 2015 6:58 PM EDT**. You got a score of **0.40** out of **3.00**. You can attempt again, if you'd like.

```
To specify an array or sequence of values in an answer, separate the value
s in
the sequence by whitespace. For example, if the question asks for the firs
t
ten powers of two (starting at 1), then the following answer is acceptable
:

     1 2 4 8 16 32 64 128 256 512

If you wish to discuss a particular question and answer in the forums, ple
ase
post the entire question and answer, including the seed (which can be used
 by
the course staff to uniquely identify the question) and the explanation (w
hich
contains the correct answer).
```

Question 1

```
(seed = 944499)
Give the array that results immediately after the 7th call (and return)
from merge() when top-down mergesorting the following array of size 12:

    72 85 84 35 20 93 77 95 48 86 13 49

Your answer should be a sequence of 12 integers, separated by whitespace.
```

You entered:

```
20   35   72   84   85   93   77   95   48   86   13   49
```

Your Answer		Score	Explanation
20 35 72 84 85 93 77 95 48 86 13 49	✖	0.00	
Total		0.00 / 1.00	

Question Explanation

```
The correct answer is: 20 35 72 84 85 93 48 77 95 86 13 49

Here is the array immediately after each call to merge():

                  72 85 84 35 20 93 77 95 48 86 13 49
merge(0, 0, 1):   72 85 84 35 20 93 77 95 48 86 13 49
merge(0, 1, 2):   72 84 85 35 20 93 77 95 48 86 13 49
merge(3, 3, 4):   72 84 85 20 35 93 77 95 48 86 13 49
merge(3, 4, 5):   72 84 85 20 35 93 77 95 48 86 13 49
merge(0, 2, 5):   20 35 72 84 85 93 77 95 48 86 13 49
merge(6, 6, 7):   20 35 72 84 85 93 77 95 48 86 13 49
merge(6, 7, 8):   20 35 72 84 85 93 48 77 95 86 13 49
```

Question 2

```
(seed = 259658)
Give the array that results immediately after the 7th call (and return)
from merge() when bottom-up mergesorting the following array:

18 57 82 35 39 85 24 38 54 27

Your answer should be a sequence of 10 integers, separated by whitespace.
```

You entered:

```
18   57   82   35   39   24   85   38   27   54
```

Your Answer		Score	Explanation
18 57 82 35 39 24 85 38 27 54	✖	0.00	
Total		0.00 / 1.00	

Question Explanation

```
The correct answer is: 18 35 57 82 24 38 39 85 27 54

Here is the array immediately after each call to merge():

                 18 57 82 35 39 85 24 38 54 27
merge(0, 0, 1):  18 57 82 35 39 85 24 38 54 27
merge(2, 2, 3):  18 57 35 82 39 85 24 38 54 27
merge(4, 4, 5):  18 57 35 82 39 85 24 38 54 27
merge(6, 6, 7):  18 57 35 82 39 85 24 38 54 27
merge(8, 8, 9):  18 57 35 82 39 85 24 38 27 54
merge(0, 1, 3):  18 35 57 82 39 85 24 38 27 54
merge(4, 5, 7):  18 35 57 82 24 38 39 85 27 54
```

Question 3

```
(seed = 322334)
Which of the following statements about mergesort are true? Check all that
apply. Unless otherwise specified, assume that mergesort refers to the pure
 recursive (top-down) version of mergesort (with no optimizations), using t
he merging subroutine described in lecture.
```

Your Answer		Score	Explanation
□ When mergesorting an array of N keys, the number of calls to merge() is ~ N. Recall	✖	0.00	The number of calls to merge() is exactly N-1. Initially, there are N subarrays that need to be merged. Each call to merge() reduces the number of subarrays by 1.

that merge() is called only on subarrays of length 2 or more.			
☐ Suppose we have a sorting algorithm that in addition to regular compares, is also allowed super-compares: take three keys and return those three keys in sorted order. Then, any compare-based sorting algorithm requires at least log_6 (N!) compares or super-compares (in the worst case) to sort an array of N items.	✖	0.00	Similar to the lower bound argument with 2-way compares, but now the height of the tree is at least log_6 (N!) since each node has as many as 6 children, corresponding to the 3! possible outcomes for each super-compare.
☑ The main reason for recurring on the left subarray before the right subarray in mergesort is to	✖	0.00	It does not affect correctness, stability, or the number of compares.

ensure stability.			
☑ It is possible to design a compare-based algorithm to merge three sorted arrays, each of size N, with no more than 6N compares.	✓	0.20	Maintain three indices that point to the smallest remaining key in each of the three sorted array. Repeatedly pick the smallest key among the three; add this to the output array; and update the corresponding index. Picking the smallest key takes at most 2 compares (fewer after one or two arrays have been exhausted). In fact, it is not hard to do it with no more than 5N compares.
□ Unstable sorting algorithms are not widely used in practice.	✓	0.20	Quicksort is an unstable sorting algorithm that is widely used in practice.
Total		0.40 / 1.00	

Question Explanation