Feedback — Mergesort

Help Center

You submitted this quiz on **Mon 21 Sep 2015 6:42 PM EDT**. You got a score of **1.43** 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 = 520354)
Give the array that results immediately after the 7th call (and return)
from merge() when top-down mergesorting the following array of size 12:

    74 77 61 93 30 86 60 56 72 65 88 82

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

You entered:

```
30  61  74  77  86  93  60  56  72  65  88  82
```

Your Answer		Score	Explanation
30 61 74 77 86 93 60 56 72 65 88 82	✖	0.00	
Total		0.00 / 1.00	

Question Explanation

```
The correct answer is: 30 61 74 77 86 93 56 60 72 65 88 82

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

                74 77 61 93 30 86 60 56 72 65 88 82
merge(0, 0, 1):  74 77 61 93 30 86 60 56 72 65 88 82
merge(0, 1, 2):  61 74 77 93 30 86 60 56 72 65 88 82
merge(3, 3, 4):  61 74 77 30 93 86 60 56 72 65 88 82
merge(3, 4, 5):  61 74 77 30 86 93 60 56 72 65 88 82
merge(0, 2, 5):  30 61 74 77 86 93 60 56 72 65 88 82
merge(6, 6, 7):  30 61 74 77 86 93 56 60 72 65 88 82
merge(6, 7, 8):  30 61 74 77 86 93 56 60 72 65 88 82
```

Question 2

```
(seed = 295580)
The column on the left contains an input array of 12 strings to be sorted;
the column
on the right contains the strings in sorted order; each of the other 4 colu
mns contains the
array at some intermediate step during either top-down or bottom-up mergeso
rt
(with different columns potentially corresponding to different algorithms).

    flax   flax   flax   flax   flax   bone
    lime   lime   jade   lime   jade   ceil
    silk   silk   lime   silk   leaf   flax
    wine   jade   mist   wine   lime   gold
```

```
   mist   mist   silk   jade   mist   jade
   jade   wine   wine   leaf   sand   leaf
   sand   sand   gold   mist   silk   lime
   leaf   leaf   leaf   sand   wine   mist
   gold   gold   sand   bone   bone   onyx
   bone   bone   bone   ceil   ceil   sand
   ceil   ceil   ceil   gold   gold   silk
   onyx   onyx   onyx   onyx   onyx   wine
   ----   ----   ----   ----   ----   ----
    0      ?      ?      ?      ?      3
```

Match up each column with the corresponding mergesorting algorithm from the given list:

0. Original input
1. Top-down Mergesort (standard recursive version)
2. Bottom-up Mergesort (nonrecursive version)
3. Sorted

You should use each choice at least once. Your answer should be a sequence of 6 integers
between 0 and 3 (starting with 0 and ending with 3), separated by whitespace.

Hint: think about algorithm invariants. Do not trace code.

You entered:

0 2 1 2 2 3

Your Answer		Score	Explanation
0	✔	0.17	
2	✖	0.00	
1	✔	0.17	
2	✔	0.17	
2	✔	0.17	
3	✔	0.17	

Total	0.83 / 1.00

Question Explanation

```
The correct answer is: 0 1 1 2 2 3

 0: Original input
 1: Top-down mergesort after 4 iterations
 1: Top-down mergesort after 9 iterations
 2: Bottom-up mergesort after forming subarrays of length 4
 2: Bottom-up mergesort after forming subarrays of length 8
 3: Sorted
```

Question 3

```
(seed = 395020)
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
☑ `Mergesort is faster in practice than insertion sort regardless of the number of items N in the array.`	✖	0.00	Insertion sort is faster for small values of N; this explains why we can improve mergesort by cutting off to insertion sort for small values of N.
☑ `The number of compares in bottom-up mergesort depends only on the size of the array N (and`	✖	0.00	The number of compares ranges from ~ 1/2 N lg N (sorted array) to ~ N lg N (random array).

not on the items in the array).			
☑ When merging two subarrays, the main reason for taking equal keys from the left subarray before the right subarray is to ensure stability.	✔	0.20	This is precisely the reason.
☐ Bottom-up mergesort uses only a constant amount of space (other than the input array).	✔	0.20	It uses linear extra space for the auxiliary array.
☐ For any array of N distinct keys, top-down mergesort and bottom-up mergesort compare exactly the same pairs of keys (but possibly in a different order).	✔	0.20	The compares can be different if N is not a power of 2. For example, consider the array 0 1 2 3 4. Top-down mergesort makes the compares { 0-1, 0-2, 1-2, 3-4, 0-3, 1-3, 2-3 } while bottom-up mergesort makes the compares { 0-1, 2-3, 0-2, 1-2, 0-4, 1-4, 2-4, 3-4 }.
Total		0.60 / 1.00	

Question Explanation