

ADS 2023 spring Week 10 Exercises

Exercises for Algorithms and Data Structures at ITU. The exercises are from *Algorithms, 4th Edition* by Robert Sedgwick and Kevin Wayne unless otherwise specified. Color-coding of difficulty level and alterations to the exercises (if any) are made by the teachers of the ADS course at ITU.

2.3.2 - Green Show, in the style of the quicksort trace given in SW 2.3, how quicksort sorts the array E A S Y Q U E S T I O N (for the purposes of this exercise, ignore the initial shuffle).

2.3.8 - Green About how many compares will `Quick.sort()` make when sorting an array of N items that are all equal?

5.1.2 - Green Give a trace for LSD string sort for the following keys:
no is th ti fo al go pe to co to th ai of th pa

5.1.3 - Green Give a trace for MSD string sort for the following keys:
no is th ti fo al go pe to co to th ai of th pa

2.3.5 - Green Describe how you would sort an array that is known to consist of items having just two distinct keys in linear time.

2.3.3 - Yellow What is the maximum number of times during the execution of `Quick.sort()` that the largest item can be exchanged, for an array of length N ?

2.3.4 - Yellow Suppose that the initial random shuffle is omitted. Give six differently ordered arrays of ten elements for which `Quick.sort()` uses the worst-case number of compares.

2.3.13 - Yellow What is the recursive depth of quicksort, in the best, worst, and average cases?

5.1.17 - Yellow Develop a version of key-indexed counting that uses only a constant amount of extra space. Prove (or argue) that your version is stable or provide a counterexample.

Old exam set 120531: 3(d-j) - Yellow See question 3 (subsections d through j) in the exam set `bads-120531.pdf` on learnit.

2.3.17 - Red Sentinels. Modify the code in Algorithm 2.5 to remove both bounds checks in the inner while loops. The test against the left end of the subarray is redundant since the partitioning item acts as a sentinel (v is never less than $a[lo]$). To enable removal of the other test, put an item whose key is the largest in the whole array into $a[length-1]$ just after the shuffle. This item will never move (except possibly to be swapped with an item having the same key) and will serve as a sentinel in all subarrays involving the end of the array. Note: When sorting interior subarrays, the leftmost entry in the subarray to the right serves as a sentinel for the right end of the subarray.

2.3.15 - Red You have a mixed pile of N nuts and N bolts and need to quickly find the corresponding pairs of nuts and bolts. Each nut matches exactly one bolt, and each bolt matches exactly one nut. By fitting a nut and bolt together, you can see which is bigger, but it is not possible to directly compare two nuts or two bolts. Describe an efficient method for solving the problem.