

Problem Set 7

This problem set is due **at 9:00pm on Wednesday, April 25, 2012.**

Both exercises and problems should be solved, but *only the problems* should be turned in. Exercises are intended to help you master the course material. Even though you should not turn in the exercise solutions, you are responsible for material covered by the exercises.

Mark the top of the first page of your solution with your name, the course number, the problem number, your recitation section, the date, and the names of any students with whom you collaborated. The homework template (\LaTeX) is available on the course website.

You will often be called upon to “give an algorithm” to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of the essay should provide the following:

1. A description of the algorithm in English and, if helpful, pseudo-code.
2. At least one worked example or diagram to show more precisely how your algorithm works.
3. A proof (or indication) of the correctness of the algorithm.
4. An analysis of the running time of the algorithm.

Remember, your goal is to communicate. Full credit will be given only to correct solutions *that are described clearly*. Convolved and opaque descriptions will receive lower marks.

Exercise 7-1. Do Exercise 20.1-1 on page 535 of CLRS.

Exercise 7-2. Do Exercise 20.1-2 on page 536 of CLRS.

Exercise 7-3. Do Exercise 20.1-4 on page 536 of CLRS.

Exercise 7-4. Do Exercise 20.3-1 on page 556 of CLRS.

Exercise 7-5. Do Exercise 20.3-2 on page 556 of CLRS.

Exercise 7-6. Do Exercise 20.3-3 on page 556 of CLRS.

Exercise 7-7. Do Exercise 20.3-4 on page 556 of CLRS.

Exercise 7-8. Do Exercise 20.3-5 on page 556 of CLRS.

Exercise 7-9. Do Exercise 20.3-6 on page 556 of CLRS.

Problem 7-1. Seeing the Forest for the van Emde Boas Trees

After listening to a lecture on van Emde Boas Trees, Ben Bitdiddle decides that he can simplify the implementation by not storing the max and min fields in each of the vEB substructures separately. Instead, he decides to insert them into the tree, while still keeping track of the global min in a separate variable. He dubs this revised version of the data structure the “Bib Tree”.

- (a) Write pseudo-code implementations of the BIB-INSERT, BIB-DELETE, and BIB-MIN functions for the Bib Tree.
- (b) Write the recurrences for the run-times of the operations in the revised data structure and solve the recurrences.
- (c) Ben Bitdiddle decides to use the Bib-Tree data structure in his implementation of Prim’s algorithm (§23.2). You can assume that the input graphs of interest all have integer weights in the range from 1 to n and that edge weights are allowed to repeat. Show how Ben would make use of his Bib-Tree and provide an analysis of time complexity for this implementation of Prim.

MIT OpenCourseWare
<http://ocw.mit.edu>

6.046J / 18.410J Design and Analysis of Algorithms
Spring 2012

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.