CS5200 Spring 2018 Homework 6

Due Tuesday, May 01,2018, 11:59 PM

Xiongming Dai

Submit your HW as follows: ontaining the following:

1. A PDF file that contains all the answers to the individual questions, all pictures, all code, and all code output. This should all be well-organized and attractively laid out. This file will contain all the grading feedback after your work is graded.
2. A ZIP file for all Python programs and input files. Also include any other files that you want considered in this Zip file.

Problems

All the problems are from the Cormen, Leiserson, Rivest, and Stein book.

1. (5 points) Do Problem 13.3-2 on p. 322.

13.3-2

Show the red-black trees that result after successively inserting the keys $41, 38, 31, 12, 19, 8$ into an initially empty red-black tree.

2. (5 points) Do Problem 13.4-3 on p. 330.

13.4-3

In Exercise 13.3-2, you found the red-black tree that results from successively inserting the keys $41, 38, 31, 12, 19, 8$ into an initially empty tree. Now show the red-black trees that result from the successive deletion of the keys in the order $8, 12, 19, 31, 38, 41$.

3. (15 points) Do Problem 15.2-2 on p. 378. Implement your algorithm and submit some sample runs to verify that it is correct.

15.2-2

Give a recursive algorithm MATRIX-CHAIN-MULTIPLY(A, s, i, j) that actually performs the optimal matrix-chain multiplication, given the sequence of matrices $\langle A_1, A_2, \ldots, A_n \rangle$, the s table computed by MATRIX-CHAIN-ORDER, and the indices i and j. (The initial call would be MATRIX-CHAIN-MULTIPLY$(A, s, 1, n)$.)

4. (5 points) Do Problem 15.4-1 on p. 396.

15.4-1

Determine an LCS of $\langle 1, 0, 0, 1, 0, 1, 0, 1 \rangle$ and $\langle 0, 1, 0, 1, 1, 0, 1, 1, 0 \rangle$.

5. (5 points) Do Problem 15.4-5 on p. 397. Just describe the algorithm. You do not need to code it.

15.4-5
Give an $O(n^2)$-time algorithm to find the longest monotonically increasing subsequence of a sequence of n numbers.

6. (5 points) Do Problem 16.1-3 on p. 422.

16.1-3
Not just any greedy approach to the activity-selection problem produces a maximum-size set of mutually compatible activities. Give an example to show that the approach of selecting the activity of least duration from among those that are compatible with previously selected activities does not work. Do the same for the approaches of always selecting the compatible activity that overlaps the fewest other remaining activities and always selecting the compatible remaining activity with the earliest start time.

7. (10 points) Do Problem 16.2-5 on p. 428. Just describe the algorithm. You do not need to code it.

16.2-5
Describe an efficient algorithm that, given a set $\{x_1, x_2, \ldots, x_n\}$ of points on the real line, determines the smallest set of unit-length closed intervals that contains all of the given points. Argue that your algorithm is correct.

8. (5 points) Do Problem 16.3-3 on p. 436.

16.3-3
What is an optimal Huffman code for the following set of frequencies, based on the first 8 Fibonacci numbers?

`a:1 b:1 c:2 d:3 e:5 f:8 g:13 h:21`

Can you generalize your answer to find the optimal code when the frequencies are the first n Fibonacci numbers?

9. (10 points) Do Problem 17.4-3 on p. 471.

17.4-3
Suppose that instead of contracting a table by halving its size when its load factor drops below $1/4$, we contract it by multiplying its size by $2/3$ when its load factor drops below $1/3$. Using the potential function

$$\Phi(T) = |2 \cdot T.num - T.size| ,$$

show that the amortized cost of a TABLE-DELETE that uses this strategy is bounded above by a constant.

10. (5 points) Do Problem 18.2-1 on p. 497.

18.2-1
Show the results of inserting the keys

$F, S, Q, K, C, L, H, T, V, W, M, R, N, P, A, B, X, Y, D, Z, E$

in order into an empty B-tree with minimum degree 2. Draw only the configurations of the tree just before some node must split, and also draw the final configuration.

11. (5 points) Do Problem 18.3-1 on p. 350.

Page 502 not page 350

18.3-1
Show the results of deleting C, P, and V, in order, from the tree of Figure 18.8(f).

12. (5 points) Do Problem 19.2-1 on p. 518.

19.2-1
Show the Fibonacci heap that results from calling FIB-HEAP-EXTRACT-MIN on the Fibonacci heap shown in Figure 19.4(m).

13. (5 points) Do Problem 19.4-1 on p. 526.

19.4-1
Professor Pinocchio claims that the height of an n-node Fibonacci heap is $O(\lg n)$. Show that the professor is mistaken by exhibiting, for any positive integer n, a sequence of Fibonacci-heap operations that creates a Fibonacci heap consisting of just one tree that is a linear chain of n nodes.

14. (15 points) Do Problem 21.3-2 on p. 572. Implement your algorithm and submit some sample runs to verify that it is correct.

21.3-2
Write a nonrecursive version of FIND-SET with path compression.

<end>