CSPP 55001 Algorithms — Autumn 2009

Homework 1 (assigned September 30, due October 7)

Reading: CLRS chapters 1 and 2; chapter 4, sections 4.1; 4.3–4.4. Reading for next week's lecture: chapters 6 and 7. Supplementary reading: chapter 5. (Second Edition, chapters 1 and 2; chapter 4, sections 4.1–4.2.)

Written assignment: Solve the following "Do" exercises and assigned problems. **Only solutions to the assigned problems should be turned in.**
Note: You are responsible for the material covered in **both** "Do" exercises and assigned problems.

"Do" Exercises (*not* to be handed in):

1. Exercise 2.3-2 on page 37.
   2nd Edition, Exercise 2.3-2, page 36.
2. Exercise 2.3-5 on page 39.
   2nd Edition, Exercise 2.3-5, page 37.
3. Prove that **mergesort** is correct. Code for **mergesort** is on page 34.
   2nd Edition, **mergesort** code is on page 32.
4. Problem 2-1, parts a–d, on pages 39–40.
   2nd Edition, Problem 2-1, parts a–d, pages 37–38.
5. Problem 2-4, parts a–d, on pages 40–42.
   2nd Edition, Problem 2-4, parts a–d, pages 39–40.
6. An array $A[1..n]$ contains all the integers from 0 to n except one. It would be easy to determine the missing integer in $O(n)$ time by using an auxilary array $B[0..n]$ to record which numbers appear in A. In this problem, however, we cannot access an entire integer in A with a single operation. The elements of A are represented in binary, and the only operation we can use to access them is "fetch the jth bit of $A[i]$", which takes constant time. Find the missing integer in $O(n)$ time using only that operation.
   2nd Edition: Problem 4-2 on page 85.

Problems (to be handed in):

1. Consider the following code for the sorting algorithm **bubblesort**.

   bubblesort(A[1..n])
   //Input: An array A[1..n] of real numbers
   //Output: Array A[1..n] sorted in ascending order
   01 **for** i ← 1 **to** n − 1
   02   **do for** j ← 1 **to** n − i
   03   **do if** A[j+1] < A[j] **exchange** A[j] **and** A[j+1]

   Let A' denote the output of bubblesort on input array A. To prove that bubblesort is correct we need to prove that it terminates and that
   $$A'[1] \le A'[2] \le \ldots A'[n],$$
   where n is the number of entries in the input array A.
   1. State precisely a loop invariant for the **for** loop in lines 2–3 and prove that this loop invariant holds. Use the structure of the loop invariant proof presented in CLRS chapter 2. (5 points)
   2. Using the termination condition of your proof in part (1), state a loop invariant for the **for** loop

in lines 1–3 and use it to prove that bubblesort is correct, i.e., that the inequality $A'[1] \leq A'[2] \leq \ldots A'[n]$ holds. (5 points)

3. What is the worst-case running time of bubblesort? How does it compare to the worst-case running time of insertion sort? (5 points)

2. Problem 4-2, parts a and b, on page 107. (5 points for each part)
   2nd Edition, Problem 4-3, pages 85–86.

3. Problem 4-5, parts a–c, on pages 109–110. (5 points for each part)
   2nd Edition, Problem 4-6, pages 87–88.

4. Let $A[1..n]$ be an array of n integers, not necessarily sorted, and let z be another given integer. Describe an algorithm that determines if, for some i, $1 \leq i \leq n$, and for some j, $1 \leq j \leq n$, $A[i] + A[j] = z$. For full credit, your algorithm should run in $O(n \log_2 n)$ time.

   Clearly describe the steps of your algorithm in English and give pseudocode, if it is helpful. Justify the running time of your algorithm and explain briefly why your algorithm is correct. You do not need to prove correctness. (10 points)

Gerry Brady
Monday October 5 19:51:13 CDT 2009