Homework 6

Algorithms and Data Structures (CH08-320201)

Spring 2018

Prof. Dr. Michael Sedlmair
Computer Science & Electrical Engineering
Jacobs University

Due on Monday, Mar 19, 2018, 23:55.

Problem 1: Sorting in Linear Time *(3+3+4+6+2+4 = 22 points)*

(a) Given the sequence $< 9, 1, 6, 7, 6, 2, 1 >$, sort the sequence by Counting Sort.

(b) Given the sequence $< 0.9, 0.1, 0.6, 0.7, 0.6, 0.2, 0.1 >$, sort the sequence by Bucket Sort.

(c) Given n integers in the range 0 to k, design an algorithm (only pseudocode) with pre-processing time $\Theta(n + k)$ that counts in $O(1)$ how many of the integers fall into the interval $[a, b]$.

(d) Given a sequence of n English words of length k, *implement* an algorithm that sorts them in $\Theta(n)$. Let k and n be flexible, i.e., automatically determined when reading the input sequence.

(e) Given any input sequence of length n, determine the worst-case time complexity for Bucket Sort. Explain your answer!

(f) Given n 2D points that are uniformly randomly distributed within the unit circle, design an algorithm (only pseudocode) that sorts the points by increasing Euclidean distance to the circle's origin.

Problem 2: Radix Sort *(8+4+2* = 12+2* points))*

Consider Hollerith's version of the Radix Sort, i.e., a Radix Sort that starts with the most significant bit and propagates iteratively to the least significant bit (instead of vice versa).

(a) *Implement* Hollerith's version of the Radix Sort.

(b) Derive the asymptotic time complexity and the asymptotic storage space required for your implementation.

(c) * Show how to sort n integers in the range 0 to $n^3 - 1$ in $O(n)$ time.

Remarks

- Solutions have to be handed in via Moodle by the due date. For late submissions you need to get in contact with the TAs directly. You need to upload one zip-file that contains a PDF-file for the theoretical parts and source files (no executables or object files) for the programming assignments. The source files need to include a makefile. Programming assignments need to be handed in as C, C++, or Python code. **Please write your own code.** It is ok to take snippets from online resources, but they need to be clearly marked.
- Exercises marked with a * are bonus problems. These count towards your number of points for this homework. The maximum number of official points can not be exceeded.