Design and Analysis of Algorithms
Homework 4

Clearly number your solution to each problem. Staple your solutions and bring them to class on the due date. Express your algorithms in pseudo-code when directed. Always provide justification for your answer when asked to give the running time of an algorithm. Be brief and concise, and draw pictures where appropriate.

Letter	A	B	C	D	E	F	G	H	I	J	K	L	M
Key	0	1	2	3	4	5	6	7	8	9	10	11	12
Letter	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Key	13	14	15	16	17	18	19	20	21	22	23	24	25

1. (a) (5pts) Give the contents of the hash table that results when the following keys are inserted in that order into an initially empty 13-item hash table: $(E_1, A, S_1, Y, Q, U, E_2, S_2, T, I, O, N)$. Use $h(k) = k \mod 13$ for the hash function for the k-th letter of the alphabet (see above table for converting letter keys to integer values). Use linear probing.

   (b) (5pts) Give the contents of the hash table that results when the same keys are inserted in that order into an initially empty 13-item hash table. Use $h(k) = k \mod 13$ for the hash function for the k-th letter of the alphabet (see above table for converting letter keys to integer values). Use double hashing and let $h'(k) = 1 + (k \mod 11)$ be the secondary hash function.

2. (a) (5pts) Draw the merge-sort tree for an execution of the merge-sort algorithm on the input sequence: (2, 5, 16, 4, 10, 23, 39, 18, 26, 15) (like in Figure. 4.2).

   (b) (5pts) Draw the quick-sort tree for an execution of the quick-sort algorithm on the input sequence from part (a) (like in Figure 4.12). Use the last element as the pivot.

   (c) (5pts) Draw the quick-sort tree for an execution of the quick-sort algorithm on the input sequence from part (a) (like in Figure 4.12). Use the element at rank $\lfloor \frac{n}{2} \rfloor$ as the pivot.

   (d) (2pts) What is the running time of the version of quick-sort in part (c) on a sequence that is already sorted? Explain.

3. (a) (5pts) Suppose we are given a sequence S of n elements, each of which is colored red or blue. Assuming S is represented by an array, give a linear-time **in-place** algorithm for ordering S so that all the blue elements are listed before all the red elements. What is the running time of your method?

   (b) (5pts) Let A and B be two sequences of n integers each. Give an integer m, describe an $O(n \log n)$ time algorithm for determining if there is an integer a in A and an integer b in B such that $m = a + b$.

4. (a) (5pts) Suppose we are given a sequence S of n elements, each of which is an integer in the range $[0, n^2 - 1]$. Describe a simple method for sorting S in $O(n)$ time. [*Hint*: think of alternative ways of viewing the elements].

   (b) (3pts) Does the running time of radix-sort depend on the order of keys in the input? Explain.

5. (5pts) A forensic lab receives a delivery of n samples. They look identical, but in fact, some of them have a different chemical composition. There is a device that can be applied to two samples and tells whether they are different or not. It is known in advance that most of the samples (more then 50%) are identical. Find one of those identical samples making no more than n comparisons. (Beware: it is possible that two samples are identical but do not belong to the majority of identical samples.)