Homework 5

Due Monday, June 14, at 9 AM (GMT-7)

CSc 345 – Summer 2014

Instructor: Qiyam Tung

Instructions

1. This is an individual assignment. You must do your own work.
2. If you are having difficulty and need to ask a question you can:
   (a) Ask questions in class.
   (b) Stop by my office hours (or make an appointment).
   (c) Post a question on Piazza.
   (d) Post a private question on Piazza if the question is too specific.
3. Show all work. Incomplete solutions will **not** receive full credit
4. You may write your solutions by hand, or you may type them using any appropriate program such as Microsoft Word, OpenOffice Writer, LaTeX, etc...
   However, the final copy should be in PDF form and formatted so that it is legible.
5. If the listed problem is only a number, refer to the online book for the description of the problem (starting at page 46).

Problems (55 points possible)

1. (20) External Sorting. Consider this sequence of keys, in the order shown:

   7, 109, 131, 53, 173, 73, 61, 43, 101, 157, 137, 3, 67, 97, 37, 113, 103, 29, 149,
   47, 197, 223, 167, 17, 2, 181, 79, 83, 151, 71, 23, 229, 191, 227, 89, 193, 31, 11,
   211, 179, 5, 59, 107, 41, 13, 163, 139, 127, 19, 199

   Use each of the following sorts to sort this list:

   - (10) 3-way external merge sort with 3 input files and 3 output files (so that the input and output roles can switch for each pass). At the end of each pass, show the content of the output files.
   - (10) Improved External Merge Sort (assume that we have four buffers, each with room for four values). Clearly indicate the content of the runs at the end of each pass, including pass 0.

2. (5) Consider the basic Quicksort algorithm, but with the pivot always chosen to be the key value at index $\lfloor \frac{n}{2} \rfloor$, where n in the quantity of values in whichever range is being partitioned currently. Describe the kind of sequences that would force this variant of Quicksort to run in $\Theta(n^2)$ time.

3. (5 points) Linear Search
   (a) (5) Problem 9.1

4. (25 points) Hashing
   (a) (5) 9.13(a,b)
   (b) (5) Bloom Filters. Let $m = 17, h_1(x) = (k+15)\%m, h_2(x) = (4k+11)\%m$, and $h_3(x) = (7k+2)\%m$. Insert the keys 23, 7, 50, and 91 into the bit vector, and show the resulting vectors content. Then, find a key that is a false positive; that is, find a key that appears to have been inserted, but wasnt.
   (c) (5) Consider a hash table with 13 slots. Draw the content of this hash table after hashing the values 72, 61, 80, 43, 17, 91, 13, 4, 10, and 1 using the hash function $h'(k) = (3k + 5)\%13$. Assume collisions are handled using chaining.
   (d) (5) Repeat 4c using quadratic probing and this function: $h(k, i) = (h'(k) + i + 2i^2)\%13$. (Note that quadratic probing is a type of open addressing; chaining is not used.) If not all values can be stored, show the content of the table at the point of the failure, and explain the problem.

(e) (5) Rehash the content of the resulting table from 4d to a table of 18 slots using linear probing ($h(k,i) = (h'(k) + i)\%18$) where $h'(k) = (3k + 5)\%18$. Take the values in the order they are encountered in 4d's table, starting with slot 0. Then, hash into the new table any values that you were unable to store in 4d's table, and then hash the values 12 and 69. Show the table as it exists after the content from 4d's table has been rehashed, and again after 69 has been hashed.