

<http://www.cs.arizona.edu/classes/cs345/spring08/>

Homework #3

(85 points)

Due Date: April 15th, 2008, at the beginning of class

Write complete, legible answers to each of the following questions. Show your work, when appropriate, for possible partial credit. This is not a group project; do your own work. The TAs will go over the homework problems with you during the section meeting following the due date. *If you need help, remember that the TAs and I have office hours for just this eventuality, and you may post general questions to the class newsgroup.*

On the due date, at the start of class, turn in your neatly written (or, better yet, electronically-formatted) answers and turn them in to the TAs. Solutions submitted after the first five minutes of class on the due date will not be accepted. Want to be safe and submit your homework early? Feel free to do so; you can give it to me or to either TA during our office hours. I strongly encourage you to retain a copy of your submitted answers, just in case you need to refer to them before you get your graded homework back.

1. [5 points] R-2.19
2. [5] R-2.21 (Use the function $h(k, i) = (h'(k) + i + 2i^2)\%11$, where $h'(k) = (2k + 5)\%11$ (the hash function from R-2.19). If an insertion fails, tell us on which item it fails and show the current table content. If all insertions succeed, just show the final table.)
3. [5] R-2.22 (Use the function $h(k, i) = (h_1(k) + ih_2(k))\%11$, where $h_1(k) = (2k + 5)\%11$, and $h_2(k)$ is the $h'()$ given in the problem. Handle successes and failures as above.)
4. [5] Is radix sort stable? Explain your answer.
5. [5] R-4.16 (Bucket Sort is explained on p. 241, and the term “in-place” is explained on p. 248.)
6. [10] C-4.16 (Do this two ways; using sorting, and without using sorting. We’ll consider your answers to be ‘efficient’ if they run in $O(n\log_2 n)$ time or better.)
7. [20] Consider this sequence of 53 characters (ignore the spaces):

```
a G d R I 6 b A W U 3 Y h f H Q 7 1 X E L J N F i
B c g C 4 P q U 2 5 D j 8 M a Z e S 9 K y m T 4 r
s V k
```

- (a) [10] Use Improved External Merge Sort to sort the characters. Follow the ASCII sequence – digits are less than upper-case letters, which are less than lower-case letters. Assume that your buffers can hold up to three characters each, and you have four buffers available. Show your collection of sorted runs at the end of each pass of the algorithm. Also, during passes 1-n, every **fourth** time you write out the content of the output buffer (that is, after the 4th, 8th, 12th, etc. writes), show the content of the input buffers at that time (the output buffer, having just been written, will be empty). That’s how we’ll be able to tell if you’re following the algorithm correctly.
- (b) [10] Sort the same collection of characters with Polyphase Merge Sort. Assume that you have 4 ‘tapes’ at your disposal (T_1 through T_4), and that the final sorted collection is to be on T_3 . Show the initial distribution of data across the tapes (use the virtual padding idea if necessary), and show the content of each tape every **third** time a tape is emptied during the sort.

(Continued on back...)

8. [10] Insert the sequence of keys (4, 15, 21, 44, 2, 17, 49, 11, 1), in the order given, into ...
- (a) [5] ... an AVL tree
 - (b) [5] ... a Splay tree

For each, show the structure after (i) the insertion of 21, (ii) the insertion of 17, and (iii) the insertion of 1.

9. [20] Consider a BST of 4 keys with k_4 at the root. Its left child is k_2 . k_2 's left and right children are k_1 and k_3 , respectively. There are five range nodes, d_0 through d_4 ; you should be able to figure out their ranges and where they go within the tree. Let $P = \langle 0.10, 0.15, 0.10, 0.15 \rangle$ and $Q = \langle 0.10, 0.10, 0.15, 0.10, 0.05 \rangle$.
- (a) [5] What is the AST of this specific tree?
 - (b) [10] What is the minimal AST over all possible trees using the same nodes and probabilities? Use the algorithm handed out in class.
 - (c) [5] Draw the optimal BST (that is, the one that possesses the minimal AST). The `root []` structure from the algorithm should prove useful for finding the tree's structure.