THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

COMPSCI.220.C.S1 – Algorithms and Data Structures

Assignment 3 – Data searching efficiency

Out: Friday, 15th of April, 2016

Due: Friday, 6th of May, 2016

Assignment 3 (Lectures 13 – 17) is worth **70 marks** representing **7%** of your total course grade.

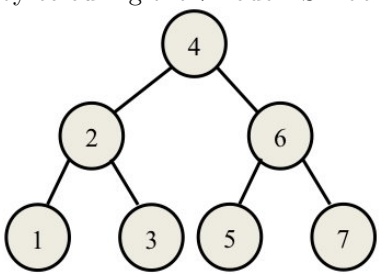
Objectives. Learning data search tools (trees and hashing) and how to analyse their efficiency.

Requirements. You should give clear and detailed answers to the following questions:

1. (*15 marks*) Suppose the seven keys 1, 2, 3, 4, 5, 6, and 7 are inserted in some order into an initially empty binary search tree (BST). Specify four insertion orders yielding trees of the maximum height and four insertion orders yielding trees of the minimum height.

2. (*5 marks*) Form an array of each unique digit from your UID (e.g., 1, 2, 3, and 6 for the exemplary UID 6232122). Form another array by taking the complement to 9 (9 minus digit) for each digit (8, 7, 6 and 3 respectively for the same example). Combine these into one array with no duplicates (1,2,3,6,7,8 for the same example) and specify two orders of their insertion into an initially empty BST that yield the minimum tree height.

3. (*15 marks*) Decide how many black nodes can be along every path from the root node to a leaf in a red-black tree built by colouring the 7-node BST below:



   List or draw all these red-black trees and indicate, which of them are the AA-trees (format for listing each tree: $[1(c_1), 2(c_2), 3(c_3), 4(c_2), 2(c_4), 5(c_5), 6(c_6), 7(c_7)]$ where $c_i \in \{\mathrm{b}, \mathrm{r}\}$ indicates black or red colour of the node $i \in \{1, 2, \ldots, 7\}$).

4. (*10 marks*) Using the hash function $h(k) = (2k + 1) \bmod 10$ and open addressing with linear probing (OALP), place to the hash table of size $m = 10$ the following seven keys: 13, 24, 43, 79, X, Y, and Z where X, Y, and Z are the first three two-digit numbers in your UID (e.g., $X = 12$, $Y = 34$, and $Z = 56$ for the exemplary UIDs 1234567 and 123456789). The answer should be in a tabular form, being similar to Table 3.3 in Textbook with the same columns "Key k", "Hash $h(k)$", "Table address", and "Comments".

5. (*10 marks*) Using the same hash function, $h(k)$, and the same seven keys, as in Question 4, fill the hash table of size $m = 10$ by open addressing with double hashing (OADH) that probes the table locations with decrement $\Delta(k) = \max\{1,\ (h(k) + k) \bmod 10\}$. The answer should be in a tabular form, similar to Table 3.4 in Textbook with the same columns "Key k", "Hash $h(k)$", "Table address", and "Comments" showing the decrements.

   Note that the chosen for simplicity of computations table size $m = 10$ is not a prime number, Thus when the decrement is equal to divisors of 10 ($\Delta(k) = 2$ or 5), the probing may encounter an infinite loop through the already occupied locations. In such a case, you should indicate the loop in "Comments" for that key and continue with inserting your next key to the table.

6. (*5 marks*) Assuming each configuration of n keys in a hash table of size m is equally likely to occur, find the expected number of probes for successful, $T_{\mathrm{ss}}(\lambda)$, and unsuccessful, $T_{\mathrm{us}}(\lambda)$, search in a hash table using OALP, OADH, and separate chaining (SC) with load factors $\lambda = 0.2$, 0.8, and 0.95. The answer should be in a tabular form, being similar to Table 3.6 in Textbook (with the same columns "λ" and "Successful search: $T_{\mathrm{ss}}(\lambda)$" and "Unsuccessful search: $T_{\mathrm{us}}(\lambda)$" with the sub-columns "SC", "OALP", and "OADH").

7. (*10 marks*) Calculate the hash addresses of your UID k in a hash table of size $m = 10,000$ for the three hash functions: (1) $h(k) = k \bmod 10,000$; (2) middle-squaring function that selects the central three-digit section of your UID, adds 320 to this 3-digit number if it is less than 320, and takes then the resulting central four-digit section, and (3) truncating function that preserves only the second, third, fifth, and sixth digits of the UID.

Submission: Your report should answer in detail to Questions 1–7. The report should be submitted as a single Adobe PDF file `CS220assign3.pdf` (only this file will be marked, so check that it can be read by PCs in the departmental Computer Labs). **Scanned handwritten text documents are strictly forbidden** (even as images in a pdf file) and will not be accepted for marking. However, scanned images of hand-drawn trees or hash tables can be used to clarify your report, despite it is more professional to prepare illustrations with any graphical editor.

Submit your file electronically to Assignment Drop Box (`https://adb.auckland.ac.nz`) (**not to Canvas!**) before Friday, 6th of May 2016, 09:00 pm (ADB time). If submitted after this due date, the penalty of 10% will be before 7th of May 2016, 09:00 p.m.; then the penalty of 50% will be before 8th of May 2016, 09:00 p.m., and no submission afterwards.

Marking scheme	% of marks
Clear structure of your report and detailed explanations	up to 20
Correctness of your final answers	up to 20
Correctness of your intermediate steps in deriving these answers	up to 20
Detailing all your steps with references, if necessary, to the textbook	up to 40
	Total: up to 100