University of Moratuwa, Sri Lanka
Faculty of Engineering
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BSc Engineering, Semester 7 (2014 Intake)
Feb – Jun 2018

CS4522 Advanced Algorithms

Assignment 1 (worth 12%; due 8th June 2018)

Note the following important points before answering this assignment:

Use standard notations. All other notations and any assumptions must be clearly explained. Some questions may not be considered during grading; but students should answer all questions. Upload a PDF document. Make sure the size of the uploaded file is less than 1MB. You must include in your submission:

- the names and registration numbers of any students with whom you worked together on this assignment (As already informed, while discussions with other students is a positive aspect, plagiarism, copying, giving or receiving aid are strictly not permitted. Working together cannot extend to writing the answers).
- the list of other resources (books, websites,…) you used in preparing your answers.

---

1. [10 marks] In a B-tree with n keys, suppose that we use binary search rather than linear search within each node in B-TREE-SEARCH. Show that this change makes the B-TREE-SEARCH time O(lg n), which is independent of t, where t is the minimum degree.

2. [10 marks] Suppose you are given the five numbers 6, 3, 5, 9, 8 to be stored in a data structure. Show a valid *treap* data structure that satisfies the *max heap property* with the above five numbers as keys. State any assumptions.

3. [30 marks] In this question, you will use the website maintained by Prof. David Galles at **https://www.cs.usfca.edu/~galles/visualization/Algorithms.html** to visualize different kinds of BSTs by performing operations on them, as specified below. On each tree, you first insert a 10-integer *arithmetic sequence* a_1, a_2, …, a_{10} with common difference between terms, d=7 and the first integer to insert, a_1, is given by the last 2 digits of your UoM registration number. For e.g., for the registration number "140011X", the last 2 digits are "11", so a_1=11; the 10-integer sequence will be: 11, 18, 25, 32, 39, 46, 53, 60, 67, 74 where a_5=39, a_{10}=74.

(a). Do the following on an initially empty BST and show the resulting tree (screen-shot) after completing each.

(i). Insert the 10-integer sequence as keys, in that order.

(ii). On the tree resulting from (i), delete the following 3 keys, in that order: smallest (a_1), fifth in the sorted order (a_5) and the largest (a_{10}).

(iii). On the tree resulting from (ii), insert 3 integers as follows as keys, in that order: a_{10}+1, a_5-1 and a_1+1

(b). Repeat (a) above for an initially empty AVL-tree.

(c). Repeat (a) above for an initially empty red-black-tree.

(d). Repeat (a) above for an initially empty splay-tree. After that, perform the following additional operations on the resulting splay-tree and show the resulting tree (screen-shot) after completing each.

(iv). Search for the smallest key in the tree.

(v). On the tree resulting from (iv), search for the largest key.

(vi). On the tree resulting from (v), search for a non-existent key close to the median key in the tree (e.g., a_6-2).

(e). Summarize your important observations about BSTs and their "balanced" forms. Obviously you are encouraged to try out much more than you are asked to do above and use all of them to write your summary. Include any critical comments you may have about the visualization tool (e.g., how it may be improved).

4. [10 marks] Answer the following questions (a)-(c) that are based on BDDs.:

(a) [3 marks] Fig Q4 shows a BDD in which two nodes representing functions *f* and *g* have been identified.

Express the two functions *f* and *g* in-terms of variables $x_1, \ldots, x_4$.

(b) [5 marks] Give an alternative BDD for *f* that has a fewer number of nodes than the BDD in Fig Q4.

(c) [2 marks] Describe “the variable ordering problem in BDDs” using an example.

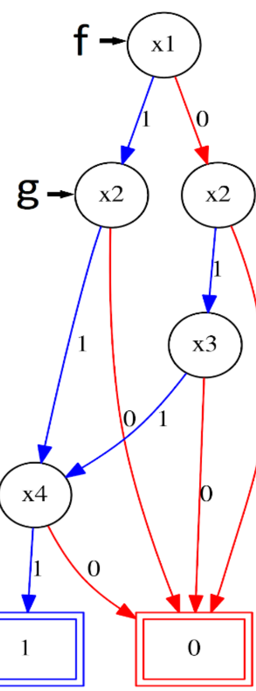

Fig. Q4

5. [40 marks] In this question you are required to run Java code for different binary search trees, namely, basic BST, red-black tree and splay tree, and report the time taken to perform basic operations on them. The Java code for each tree can be found in the "code" folder provided with this assignment.

Note that the provided code for each tree implements all the tree operations you need, but its main(_) method is empty. For each type of tree, you need to fill in the code in the main( ) method to do necessary initializations, read data from a file into an initial empty tree and then perform various tree operations, as described below in (a)-(c). You are also provided a zip file "data.zip" that contains two sets of data, each containing 3 data files (to be used as inputs to the program).

(a). Implement and measure the time taken to **insert** data in each data file to each search tree. Report the time taken, using the table structure shown below. After presenting the results, you should discuss about them, addressing important observations and explaining key points (use charts if needed).

Data Set		Time taken to **Insert** (in Microseconds)		
		Basic BST	Splay Tree	RB-Tree
Set 1	data_1			
	data_2			
	data_3			

Data Set		Time taken to **Insert** (in Microseconds)		
		Basic BST	Splay Tree	RB-Tree
Set 2	data_1			
	data_2			
	data_3			

(b). Similarly, perform **search** operation, where the data to be searched are specified in the files provided in the “search” folder for each tree. Report the time taken, using the table structure shown below. Explain the behaviour of this operation across different types of search trees.

Data Set		Time taken to **Search** (in Microseconds)		
		Basic BST	Splay Tree	RB-Tree
Set 1	data_1			
	data_2			
	data_3			

Data Set		Time taken to **Search** (in Microseconds)		
		Basic BST	Splay Tree	RB-Tree
Set 2	data_1			
	data_2			
	data_3			

(c). Similarly, perform **delete** operation, where the data to be deleted are specified in the files provided in the “delete” folder for each tree. Report the time taken, using the table structure shown below. Explain the behaviour of this operation across different types of search trees.

Data Set	Time taken to **Delete** (in Microseconds)		
	Basic BST	Splay Tree	RB-Tree

Set 1	data_1			
	data_2			
	data_3			

Data Set		Time taken to **Delete** (in Microseconds)		
		Basic BST	Splay Tree	RB-Tree
Set 2	data_1			
	data_2			
	data_3			

What to Submit: At the end, you are required to submit a zip file that contains your modified Java source files and the PDF document that contains your answers to the questions 1-5; name the submitted zip file as "A1_<UoMRegNo>.zip" where <UoMRegNo> is your UoM Registration number (e.g., 140011X).