THE UNIVERSITY OF CALGARY

FACULTY OF SCIENCE

FINAL EXAMINATION

Computer Science 331

DATE: Practice TIME: 2 hrs.

NAME: ____

Aids Allowed: One double-sided letter-sized page of notes.

Use of Electronic Devices: The use of camera devices, MP3 Players and headphones, or wireless access devices such as cell phones or Blackberries during the examination is ***not*** allowed. Calculators are ***not*** allowed for this examination.

Instructions:

- Answer questions in the space provided in this examination booklet. The last two pages may be used to continue answers if you run out of space.
- Answer ALL questions. See the more detailed grade breakdown on the next page for additional information about what happens if you answer additional questions.
- Asymptotic notation can be used to state bounds on running times — but you must use this notation correctly, and as precisely as possible.

ID: ________________________

Question		**Score**	**Out Of**
1.	Operations on Binary Search Trees		10
2.	Red-Black Trees		10
3.	Hash Tables		10
4.	Searching in Sorted Arrays		10
5.	Quick Sort		5
6.	Heaps and Heap Sort		15
7.	Graphs and Graph Algorithms		15
Total:			**75**

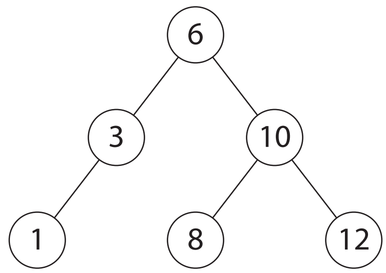
Breathe!!! *RELAX!!!!!*

Then, please do the following.

1. Read ***all*** of a question ***carefully*** before you begin to answer it. ***Ask about it*** if the instructions for a question are confusing or unclear.
2. Then do your best.

ID: ________________________

1. Consider the following ***binary search tree*** T.



(5 marks) (a) Describe, in your own words, a method to ***insert*** a value into a binary search tree. Then draw the tree that would be produced by inserting 7 into the tree T:

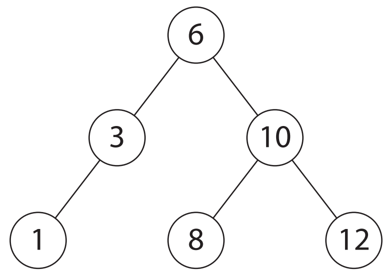
Your Description of an Insertion Algorithm:

Tree Obtained by Inserting 7 into T:

ID: ________________________

(5 marks)

(b) Suppose that you wish to ***delete*** a value that is stored at a node with two children. Describe how this should be done. Then draw the binary search tree that would be produced after deleting 6 from T:



How To Delete a Value at a Node with Two Children:

Tree Obtained by Deleting 6 from T:

ID: ________________________

2. Consider a ***red-black tree***.

(2 marks) (a) Recall that the ***height*** height(x) of a node $\mathtt{x}$ is the maximum number of nodes on a path from $\mathtt{x}$ down to a leaf (not including $\mathtt{x}$), and that the ***black-height*** black-height(x) of $\mathtt{x}$ is the number of ***black*** nodes on a path from $\mathtt{x}$ down to a leaf (not counting $\mathtt{x}$).

Explain why height$(\mathrm{x}) \leq 2 \times$ black-height(x) for every node $\mathtt{x}$ in a red-black tree.

Why the Height is Never More Than Twice the Black-Height:

(2 marks) (b) Suppose that T is a red-black tree with size n. Describe, as precisely as you can, how the ***depth*** of this tree is related to its size.

Relationship between Depth and Size for a Red-Black Tree;

ID: ________________________

(6 marks)

(c) Consider the ***deletion*** of a value from a red-black tree T when this value is in the tree. Recall that some node `y` (with at most one non-NIL child) is removed and that a child `x` of `y` gets promoted to replace it. Describe

- the **colour** given to `x` when it is promoted (and how this depends on the colour of `y`)
- the problem that this might cause, and
- (somewhat briefly and informally) how this problem is corrected.

Colour Given to `x` *if* `y` *was Red:*

Colour Given to `x` *if* `y` *was Black:*

Problem(s) That Might be Caused by This:

How This is Fixed:

ID: ________________________

3. Consider the use of ***hash tables*** to store finite sets.

(3 marks) (a) First consider ***hashing with chaining***, using table size $m = 10$ and using the following hash function to store integers:

$$h(x) = x \pmod{10}.$$

Draw the hash table that would be produced by inserting the values

$$0, 20, 10, 30, 5$$

(in this order) into an initially empty table.

Resulting Hash Table:

(3 marks) (b) Repeat the above question, using ***hashing with open addressing***. In this case you should use a hash function such that

$$h(x, 0) = x \pmod{10}$$

and such that

$$h(x, i) = x + i^2 \pmod{10}$$

for every integer i such that $1 \leq i \leq 9$ — so that ***quadratic probing*** is being used.

Resulting Hash Table:

ID: ________________________

(2 marks) (c) Next draw the hash table that you would get by starting with the hash table you produced when answering part (b) and ***deleting*** 20.

Hash Table with Open Addressing Obtained by Deleting $\mathbf{20}$*:*

(2 marks) (d) Describe briefly why it is **not** generally a good idea to use hash tables with open addressing if ***deletions*** are common.

Why You Should Avoid Hash Tables with Open Addressing When Deletions are Frequent:

ID: ______________________

4. Consider the problem of ***searching in a sorted array***.

(5 marks) (a) Describe, IN YOUR OWN WORDS, the ***binary search*** algorithm. Recall that this algorithm

- accesses as global data, but does not change, an array `A` that has positive length and stores elements of an ordered type `T` in nondecreasing order, as well as a key `k` of the type of values stored in the array;
- receives integers `low` and `high` such that

$$0 \leq \texttt{low} \leq \texttt{A.length} \quad \text{and} \quad -1 \leq \texttt{high} \leq \texttt{A.length} - 1$$

as input.

If $\texttt{low} \leq \texttt{high}$ and there exists an integer i such that $\texttt{low} \leq i \leq \texttt{high}$ and $\texttt{A}[i] = \texttt{k}$, then an integer i with these properties is returned as output. A `NoSuchElementException` is thrown, otherwise.

Your Description of the Binary Search Algorithm:

(3 marks) (b) Consider the following sorted array.

0	1	2	3	4	5	6	7
1	3	5	9	16	25	30	40

List the sequence of values that the key $\texttt{k} = 30$ is compared to, during a search with this array, and with inputs $\texttt{low} = 0$ and $\texttt{high} = 7$.

Values Visited:

(2 marks) (c) Which of the algorithms — ***Binary Search*** or ***Linear Search*** — should you use if you wish to minimize the number of steps used in the worst case to search for a value in a large sorted array? Why?

Which Algorithm Should be Used — and Why:

ID: ________________________

5. Consider the ***Quick Sort*** **algorithm.**

(2 marks) (a) The deterministic version of this algorithm, presented in class, always used the *last* element in the part of the array being sorted as the ***partition element*** (or the "*pivot element*") when partitioning the array.

Assuming that this algorithm is being used to sort an array with length n, give an upper bound for the number of steps used to sort the input array, ***in the worst case***, as a function of n, that is as precise as you can.

Upper Bound for Number of Steps Used:

(3 marks) (b) Describe a way to modify the algorithm so that its asymptotic worst-case performance is improved. What is the worst-case performance of the new version of the algorithm?

Modification of the Algorithm:

New Worst-Case Performance:

ID: ________________________

6. Consider ***Heaps*** and the ***Heap Sort*** algorithm.

(7 marks)

(a) Recall that the algorithm to ***delete the largest value*** from a Maxheap begins by reading and storing the value at the ***root,*** so that it can later be returned as output. It then overwrites this value with the value at a ***leaf***. This leaf is then deleted from the Maxheap.

Unfortunately, the resulting structure is *not* generally a Maxheap.

Explain, IN YOUR OWN WORDS, ***why*** this is true and ***what the algorithm does*** to turn the resulting tree into a Maxheap, once again. Finally, state the ***number of steps executed by this algorithm,*** in the worst case, as a function of the ***size*** n of this Maxheap.

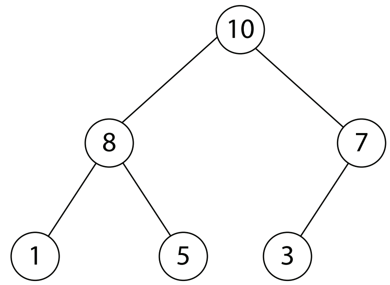
Why This is Not Generally a Maxheap:

What the Algorithm Does to Produce a Maxheap:

Number of Steps Used in the Worst Case:

ID: ____________________

Now consider the following Maxheap H:



(2 marks) (b) Fill in the entries of the ***array*** representation of this Maxheap:

Resulting MaxHeap and Heap Size:

0	1	2	3	4	5	6	7

heap-size =

(3 marks) (c) Draw the Maxheap that would be produced by deleting the maximal element from H.

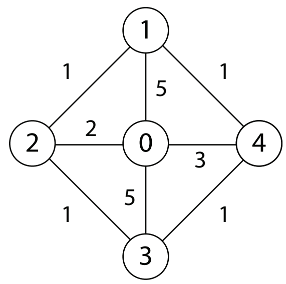
MaxHeap After Deletion:

(3 marks) (d) Describe, IN YOUR OWN WORDS, how the ***Heap Sort*** uses the `insert` and `deleteMax` operations, for a Maxheap, to sort an array.

Description of Heap Sort:

ID: ________________________

7. Consider the following ***weighted graph*** $G = (V, E)$:



(3 marks) (a) Draw the ***adjacency list representation*** of G.

Adjacency List Representation:

(3 marks) (b) Draw the ***adjacency matrix representation*** of G.

Adjacency Matrix Representation:

ID: ________________________

(2 marks) (c) State the number of steps used, in the worst case, to ***decide whether a given pair of vertices are neighbours***, using each of the above representations of G. This might depend on the size of either V or E.

Number of Steps Used with Adjacency List Representation:

Number of Steps Used with Adjacency Matrix Representation:

(3 marks) (d) Give the definition of a ***minimum cost spanning tree*** of a connected weighted graph $G = (V, E)$.

Definition of a Minimum-Cost Spanning Tree:

(3 marks) (e) Draw a minimum cost spanning tree of the graph $G = (V, E)$ shown on the previous page.

Minimum Cost Spanning Tree:

(1 mark) (f) Finally, state the number of steps used, in the worst case, to compute a minimum cost spanning tree for a weighted graph $G = (V, E)$ using ***Prim's algorithm***. You may assume that an adjacency list representation has been given for G.

Number of Steps Used by Prim's Algorithm in the Worst Case:

ID: ___________________________

ID: ______________________