Duration: **50 minutes**
Aids Allowed: **One *single*-sided *hand*written *8.5"×11"* aid sheet.**

Student Number: |_|_|_|_|_|_|_|_|_|_|_|

Last (Family) Name(s): _____

First (Given) Name(s): _____

---

*Do **not** turn this page until you have received the signal to start.*
In the meantime, please read the instructions below *carefully*.

This term test consists of 4 questions on 10 pages (including this one), printed on both sides of the paper. *When you receive the signal to start, please make sure that your copy of the test is complete, fill in the identification section above, and write your name on the back of the last page.*

Answer each question directly on the test paper, in the space provided, and use one of the "blank" pages for rough work. If you need more space for one of your solutions, use a "blank" page and *indicate clearly the part of your work that should be marked.*

In your answers, you may use without proof any theorem or result covered in lectures, tutorials, problem sets, assignments, or the textbook, as long as you give a clear statement of the result(s)/theorem(s) you are using. You must justify all other facts required for your solutions.

Write up your solutions carefully! In particular, use notation and terminology correctly and explain what you are trying to do —part marks *will* be given for showing that you know the general structure of an answer, even if your solution is incomplete.

If you are unable to answer a question (or part of a question), remember that you will get 10% of the marks for any solution that you leave *entirely blank* (or where you cross off everything you wrote to make it clear that it should not be marked).

Marking Guide

Nº 1:	_____/12
Nº 2:	_____/10
Nº 3:	_____/12
Nº 4:	_____/ 6
Bonus:	_____/ 4
TOTAL:	_____/40

Good Luck!

Question 1. [12 MARKS]

Consider the following algorithm that finds the *last* occurrence of an element x in an array A.

```
FindLast(A, x):
    i ← A.length − 1
    while i ⩾ 0 and A[i] ≠ x:
        i ← i − 1
    return i
```

For this question, we measure complexity by counting only the number of array accesses (like in Question 1 of Assignment 1).

Part (a) [2 MARKS]

What is the worst-case complexity of FindLast? Give an **exact** expression and justify your answer.

Part (b) [10 MARKS]

What is the average-case complexity of FindLast? Clearly define your sample space and probability distribution—assume that $\Pr[x \notin A] = 1/3$—explain what you are doing, show your work, and simplify your answer.

Question 1. (continued)

Part (b) (continued)

Question 2. [10 MARKS]

Recall that the *rank* of an element in a collection is its position in the sorted order of all the elements (where counting starts at 1). When duplicate values are allowed, the same value can have many different ranks. For example, in the collection $\{5.5, 1.3, 1.3, 8.9, 0.5, 3.4, 0.5\}$, the rank of 0.5 is both 1 and 2, while 3.4 has rank 5.

Consider the following "k-SET" ADT (for fixed $k \in \mathbb{Z}^+$).

Objects: A collection C of rational numbers (duplicate values are allowed).

Operations:

- INSERT(C, x): add element x to collection C.
- k-SELECT(C): return the element at rank k in C—or NIL if $|C| < k$.

Describe an implementation of the k-SET ADT, based on data structures covered in class. State the worst-case running time of each of your operations. *For full marks, your operations must be as efficient as possible—in particular, k-SELECT must run in worst-case time $O(1)$.*

Use the space on this "blank" page for scratch work, or for any solution that did not fit elsewhere.
Clearly label each such solution with the appropriate question and part number.

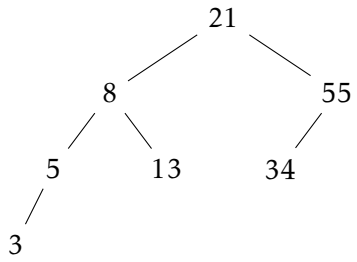
Question 3. [12 MARKS]

Part (a) [4 MARKS]

Draw an AVL tree containing items $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$, but make your tree *as unbalanced as possible*. Next to each node, indicate the height of the subtree rooted at that node.

Part (b) [4 MARKS]

Draw the result when 55 is deleted from the AVL tree below. Show your work—in particular, identify clearly any rotation performed.



Question 3. (continued)

Part (b) (continued)

Part (c) [4 marks]

Let $m(h)$ be the *minimum* number of nodes in any AVL tree of height h. Give a recurrence relation for $m(h)$, including appropriate base cases. Explain your recurrence briefly.

Question 4. [6 MARKS]

We want to keep track of student records in CSC 263 H1 (with 135 students currently enrolled). The "key" for each student will be their student number.

Part (a) [3 MARKS]

Consider using a hash table of size 11 with chaining. Assuming we use a hash function that satisfies *simple uniform hashing*, how many student records will we encounter, on average, when searching for a specific student? State clearly the general expression for your answer, then give the value for this particular example (you can give your final answer as a fraction).

Part (b) [3 MARKS]

Consider using a different hash table with size 135 and the hash function $h(N)$ = sum of the individual digits of N. (For example, $h(1234567890) = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 0 = 45$.) Is this a good hash function to use? Give at least two good reasons to support your answer.

Bonus. [4 marks]

WARNING! This question is difficult and will be marked harshly: credit will be given only for making *significant* progress toward a correct answer. Please attempt this bonus only *after* you have completed the rest of the test.

Solve your recurrence from Question 3(c).

On this page, please write nothing except your name.

Last (Family) Name(s): _____

First (Given) Name(s): _____

Total Marks = 40