York University
EECS 2011Z Winter 2018 – Midterm Tuesday Feb 27th
Instructor: James Elder

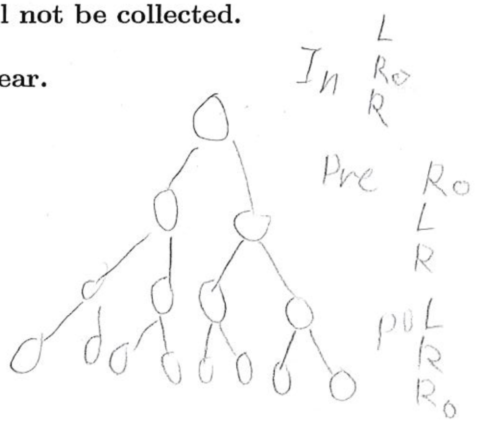
Family Name: Jae Given Name: Sangheon

Student #: 213166392 Email: kses1234@my.yorku.ca

Problem	Out of	Score
1	$6+6=12$	12
2	$5\times 2=10$	10
3	$6+6=12$	0
4	$4\times 4=16$	15
5	$20+4=24$	1
6	$13+4+3+3+3=26$	0
Total	**100**	38

Please place your York ID card on your desk for identification purposes.
This test is closed book and lasts 75 minutes: 0.75 minutes per mark.
You may not use any electronic/mechanical computation devices.
Paper for rough work can be provided but will not be collected.

Keep your answers short and clear.
Good luck!



54

440 12 $\frac{1}{n^{0.5}}$ $n^{-0.5}$ $n^2 + 2n + 1$

$n + 2 + \frac{1}{n} \leq n$

1. (6 + 6 = 12 marks) Provide tight asymptotic bounds for the following functions of n. Justify your answer as formally as possible.

(a) $f(n) = \frac{(n+1)^2}{n} = \Theta(n) = O(n)$ and $\Omega(n)$

O definition → $f(n) = O(n) = \exists c_0, n_0 > 0$ such that $\forall n \geq n_0$, $f(n) \leq c_0 n$

e.g. $c_0 = 2, n_0 = 1$, $\frac{(n+1)^2}{n} \leq 2n$

6

Ω definition $f(n) = \Omega(n) = \exists c_0, n_0 > 0$ such that $\forall n \geq n_0$, $f(n) \geq c_0 n$

e.g. $c_0 = 1, n_0 = 1$, $\frac{(n+1)^2}{n} \geq n \Leftrightarrow \frac{n^2+2n+1}{n} \geq n \Leftrightarrow n + 2 + \frac{1}{n} \geq n$

(b) $f(n) = 10n^2 + \frac{1}{\sqrt{n}}n^3$

$f(n) = 10n^2 + \frac{1}{\sqrt{n}}n^3 = 10n^2 + \frac{1}{n^{0.5}}n^3 = 10n^2 + n^{-0.5}n^3 = 10n^2 + n^{2.5}$

$\Theta(n^{2.5}) \Leftrightarrow O(n^{2.5})$ and $\Omega(n^{2.5})$

$f(n) = O(n^{2.5})$ = definition given above

6

$f(n) = \Omega(n^{2.5})$ = definition given above

P.O = primitive operation

10

2. (5 × 2 = 10 marks) Provide a tight bound on the asymptotic run time of each of the following Java code fragments, in terms of n. You do not need to justify your answer.

(a)
```
for (i = 1; i <= n; i++) {
    for (j = 1; j <= n; j++) {
        System.out.println("All work and no play makes Jack a dull boy.");
    }
}
```
Answer: $\Theta(n^2)$, because each of n outter iteration does n time of iterations, causing there will be at least n^2 times of primitive operation $(\Omega(n^2))$ and at most $c_0 n^2$ times of p.o $(O(n^2))$

(b)
```
for (i = 1; i <= n*n; i++) {
    for (j = 1; j <= n; j++) {
        System.out.println("All work and no play makes Jack a dull boy.");
    }
}
```
Answer: $\Theta(n^3)$ outter iteration do n times operation at each time, Outter iteration runs at least n^2 times operation so at least n^3 times p.o and at most $c_0 n^3$ times p.o $(\Omega(n^3))$ $(O(n^3))$

(c)
```
for (i = 1; i <= n; i++) {
    for (j = 1; j <= n*n; j++) {
        System.out.println("All work and no play makes Jack a dull boy.");
    }
}
```
Answer: $\Theta(n^3)$ It is similar case with (b), outter do n times while inner do n^2 times. Total at least n^3 $(\Omega(n^3))$ and $c_0 n^3$ at most $(O(n^3))$

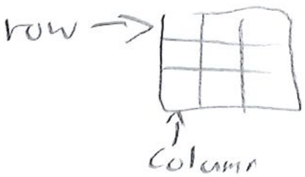
(d)
```
for (i = 1; i <= Math.sqrt(n); i++) {
    for (j = 1; j <= Math.sqrt(n); j++) {
        System.out.println("All work and no play makes Jack a dull boy.");
    }
}
```
Answer: Outer iteration $= \sqrt{n}$ times and inner iteration $= \sqrt{n}$ times Totat $(\sqrt{n})^2 = n$ times iteration so it would be $\Theta(n)$

(e)
```
for (i = 1; i <= Math.sqrt(n); i++) {
    for (j = 1; j <= n*n; j++) {
        System.out.println("All work and no play makes Jack a dull boy.");
    }
}
```
Answer: Outter $= \sqrt{n}$ times iterations

Inner $= n^2$ times iterations

Total $= \sqrt{n} \cdot n^2 = n^{2.5}$ iterations

So $\Theta(n^{2.5})$



0

3. (6 + 6 = 12 marks) You are designing a new linear algebra Java library. One of the classes in your library is called Matrix. An object of class Matrix contains a 2D rectangular array of double values.

   The signature for the Matrix constructor is shown below:

```
public Matrix (double[][] matrix) throws InvalidMatrixException
```

   (a) Please explain the conditions under which your constructor will throw an InvalidMatrixException. Note that row vectors, column vectors, scalars and null values are all considered Matrices and should not result in an exception.

   (b) Please explain in English, pseudocode or Java how you would test for these conditions.

15

4. (4×4 = 16 marks) You are designing a system that automatically selects a sequence of requested songs to play on York's student-run radio station. A requested song can be played at most once per day. Each song is represented by an entry in your data structure, and you are free to add elements to the contents of an entry (e.g., a key). Please indicate the ADT you will use to store requested songs to efficiently satisfy the daily conditions below. Justify your answer in at most two sentences.

   (a) On Mondays, you play the most recently requested song next.

   ADT = stack

   Most recent element stays on the top of stack so popping from stack will give most recent song and keep recent song by pushing

   (b) On Tuesdays, you play songs in the order they are requested.

   ADT = Queue

   Using FIFO queue, when we dequeue from queue, it will gives us the songs by order of requested

   (c) On Wednesdays, you play songs that have been requested so far that day, but in order of their ranking on the Billboard 100 charts for that week.

   ADT = Priority Queue

   Each entry of priority queue contains key (rank of billboard) and value(song) so whatever sequence you put entries in queue, it will give minimum rank song

   (d) On Thursdays, you select songs based upon how many requests they have received so far that day - the most requested song is played next. Note that requests come in continuously. You may assume that each request enters the system as a complete entry for the requested song.

   ADT = Priority Queue (with maxheap)

   Adaptable

   Set key with object implenting Comparable. Adding song in list ~~if~~ it's song already exist, then increment object and uphead, If not, add at the end of heap.

1

5. (20+4 = 24 marks) You have designed a clinical study for a new drug called Einstein that is purported to raise IQ by 30 points. To test the drug you gave it to a sample of n_t test subjects, and gave a placebo to a separate sample of n_c control subjects. The data records for the two samples of subjects are stored in two singly-linked lists named test and control, and are sorted by pre-study (baseline) IQ (a positive integer).

To perform a statistical analysis of results, you now wish to construct the largest possible subset of IQ-matched 1:1 pairings of test and control subjects. In other words, for every test subject in your subset there should be a unique control subject with matching baseline IQ, and vice-versa.

(a) Provide the pseudo-code for an iterative algorithm (testp, controlp) = pair(test, control) that efficiently extracts and returns this maximal 1:1 pairing of test and control subjects as two new singly-linked lists testp and controlp. You may assume that each list L provides an iterator over nodes L.iterator(), that an element can be accessed from a node A using A.getElement(), and that the IQ stored in an element e can be accessed using e.getIQ(). An element can be added to the end of a list L using L.add(e).

0

Prof said test.length() == control.length() == newList.length() Since test and control is sorted, and we need to have 1:1 paring for every test and control, it. simply means same index test and control is linked. It is the only case cause we need new list to be same length and 2' lists are already sorted,

```
Algorithm pair(test,control)
  newList ← new singly-linked lists with 2 elements test.head
                                                     control.head
  int a ← 1
  while ( a < test.size) do
?   newList.setNext ← new Node (newList.test.next,
                                newList.control.next)
    a++
  end while
```

(b) What is the asymptotic run time of your algorithm, in terms of n_t and n_c? Justify your answer in one or two sentences.

1

$O(n_t)$, $n_t == n_c$ and we only visit once in each element

0

6. (13 + 4 + 3 + 3 + 3 = 26 marks) Neurotree is an online project that documents academic genealogy according to mentoring relationships in the neurosciences. Every node in the tree contains an element that represents a neuroscientist, and the children of the node (the 'mentees') are the graduate students and postdoctoral fellows they have mentored. Let us assume for simplicity that each neuroscientist appears only once in the tree and that a mentee has at most one mentor. We define the mentorship distance dist(A,b) as the depth of the node B containing element b in the subtree rooted at node A. If B is not a descendent of A, dist(A,b)= ∞. The tree uses a linked structure. An iterable collection of the child nodes of node A can be accessed using A.children(), and its parent node can be accessed using A.parent(). A.element() returns the element stored at node A and a comparator compare(a,b) can be used to compare the values of two elements a and b.

   (a) Provide an efficient recursive pseudocode implementation of dist(A,b).

   (b) Provide a tight bound on the asymptotic run time of your algorithm in terms of the number n of nodes in the subtree rooted at A. Justify your answer in one or two sentences.

(c) Suppose that the signature for your algorithm is changed to be dist(A,B), where B is now the node containing mentee element b.

  i. Describe in one to two sentences how you would change your algorithm to make it more efficient.

  ii. What is the asymptotic run time of your algorithm in terms of the height h of the subtree rooted at A?

  iii. If the tree is a complete binary tree, what is the run time in terms of the number n of nodes in the subtree rooted at A?