NATIONAL UNIVERSITY OF SINGAPORE
SCHOOL OF COMPUTING

TERM TEST FOR
Semester 2, AY2019/2020

CS3230 – DESIGN AND ANALYSIS OF ALGORITHMS

7 Mar 2020 Time Allowed: 2 hours

Instructions to Candidates:

1. This paper consists of **FOUR** questions and comprises **FOURTEEN (14)** printed pages, including this page.
2. Answer **ALL** questions.
3. Write ALL your answers in this examination book.
4. This is an **OPEN BOOK** examination.

Matric. Number: ____________________
Tutorial Group Number: ____________________

QUESTION	POSSIBLE	SCORE
Q1	20	
Q2	20	
Q3	30	
Q4	20	
TOTAL	**90**	

IMPORTANT NOTE:

- *You can **freely quote** standard algorithms and data structures covered in the lectures and homeworks. Explain **any modifications** you make to them.*
- *Unless otherwise specified, you are expected to **prove (justify)** your results.*

Q1. (20 points) Sorting out order of growth rates

(a) Rank the following functions in *increasing order of growth*; that is, if function $f(n)$ is *immediately* before function $g(n)$ in your list, then it should be the case that $f(n)$ is $O(g(n))$.

$g_1(n) = \sum_{i=1}^{\lg n-1} \lg\lg \frac{n}{2^i}$ $\quad$ $g_2(n) = \sum_{i=1}^{n-2} \lg\lg(n-i)$ $\quad$ $g_3(n) = (\lg n)!$

$g_4(n) = 2^{\lg\lg\lg n}$ $\quad$ $g_5(n) = 10^{\lg((\lg\lg n)!)/\lg\lg n}$ $\quad$ $g_6(n) = \lg((\lg n)!)$

$g_7(n) = n^3$ $\quad$ $g_8(n) = 2^n$

To simplify notations, we write $f(n) << g(n)$ to mean $f(n) = o(g(n))$ and $f(n) \equiv g(n)$ to mean $f(n) = \Theta(g(n))$. For example, the four functions n^2, n, $(2013n^2 + n)$ and n^3 could be sorted in increasing order of growth as follows: ($n << n^2 \equiv (2013n^2 + n) << n^3$). *Proofs are not required for this problem.*

(b) Can you show that $n^2 = O(e^n)$ by limit method?

Q1. (continued…)

(c) Can you show that $n^2 + 3n \lg n = O(e^n)$ by definition? Please show the steps and state clearly what are c and n_0.

Q2. (20 points) Solving recurrence

(a) Among the following recurrences, can you find their time complexities? Please show the detail steps.

(i) $T(n) = 9T(n/3) + n^2 / \lg n$

(ii) $T(2^n) = T(2^0) + T(2^1) + \dots + T(2^{n-1}) + n^2$

(iii) $T(n) = 5\,T(n/4) + n^4 / \lg \lg n$

(iv) $T(n) = 4\,T(n/4) + n \lg \lg \lg n$

Q2. (continued…)

Q3. (30 points) Divide-and-conquer

You are given a set of points P={ (x_1, y_1), (x_2, y_2), …, (x_n, y_n) }. Assume the points are sorted such that $y_1 \le y_2 \le \ldots \le y_n$. We hope to find the closest pair of points. (Distance is measured using Euclidean metric.)

(a) Consider the following algorithm Closest(P) that return the distance of the closest pair of points. Can you show that the algorithm is correct?

Closest(P)

1. When P has 1 point, return ∞;
2. When P has 2 points, return their distance;
3. Find the median x_{med} of the x-coordinates of P by linear time select
4. Partition the point set P into two equal halves P_1 and P_2 by x_{med}
5. d_1=Closest(P_1);
6. d_2=Closest(P_2);
7. d=min{d_1, d_2};
8. for (x_i, y_i) in P_1,
9.   for (x_j, y_j) in P_2,
10. $d = \min\left\{d, \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}\right\}$
11. Return d;

Q3. (continued…)

(b) Let T(n) be the running time of Closest(P) when P has n points. Can you give the recursive equation for T(n)? Please also compute the time complexity of Closest(P)?

Q3. (continued…)

(c) Suppose d_1=Closest(P_1), d_2=Closest(P_2) and d=min{d_1, d_2}. Let Q_1={ (x_i, y_i)$\in P_1$ | $x_{med} - x_i \leq d$} and Q_2={ (x_i, y_i)$\in P_2$ | $x_i - x_{med} \leq d$}. Let d′ be the shortest distance among all point pairs between Q_1 and Q_2. Can you show that the shortest distance among all points in P is min{d, d′}?

Q3. (continued…)

(d) For each point (x,y) in Q_1, only points (x', y') in Q_2, where $|y'-y| \leq d$, are within distance d from (x, y). The number of such points is at most 6. Can you show that this statement is true?

Q3. (continued…)

(e) Denote $Q_1=\{(x_1, y_1), (x_2, y_2), \ldots, (x_r, y_r)\}$ and $Q2=\{(x'_1, y'_1), \ldots, (x'_s, y'_s)\}$. Assume the points in both Q_1 and Q_2 are sorted by their y-coordinates (i.e. $y_1 \leq y_2 \leq \ldots \leq y_r$ and $y'_1 \leq y'_2 \leq \ldots \leq y'_s$). Can you give an O(n)-time algorithm Shortest(Q_1, Q_2, d) that computes min{d, d′} where d′ is the shortest distance among the point pairs in $Q_1 \times Q_2$?

Q3. (continued…)

(f) Can you give an O(n log n) time algorithm to find the shortest distance among all points in P?

Q4. (20 points) Randomized Algorithm and Lower bound

(a) Consider a hash table with 100 entries. We store k arbitrary values x_1, x_2, …, x_k into the hash table. What is the minimum k so that we expect there are at least 5 pairs of values (x_i, x_j) such that $h[x_i]=h[x_j]$? (Note that the hash function h is a randomized procedure satisfying the universal hashing assumption.)

Q4. (continued…)

(b) There are 2^k balls. All balls have the same weight and the same volume except that exactly one ball is heavier and exactly one ball is bigger. (It is possible that the ball which is heavier is the same as the ball which is bigger.) You have a weight machine and a volume machine. Given two groups of balls, the weight machine reports the group which is heavier; otherwise, it reports same weight. The volume machine reports the group which has bigger volume; otherwise, it reports same volume.

i. Can you give the exact lower bound (not asymptotic bound) of the number of measurements to determine the heavy ball and the big ball?

ii. Can you give an optimal algorithm that identifies the heavy ball and the big ball?

-- End of Paper ---