NATIONAL UNIVERSITY OF SINGAPORE

CS2040 – DATA STRUCTURES AND ALGORITHMS

(Semester 2: AY2018/19)

Time Allowed: 2 Hours

INSTRUCTIONS TO STUDENTS

1. Do **NOT** open the question paper until you are told to do so.
2. This assessment paper consists of **Fourteen (14)** printed pages and **Fourteen (14)** questions with possible subsections.

   Important tips: Pace yourself! Do **not** spend too much time on one (hard) question.
3. Answer all questions. Use the provided OCR form for the 10 MCQ questions, and write your answer for the structured questions directly in the space given after each question.
4. Use 2B or darker pencil to shade the OCR form. You are allowed to use PENCIL to write your answers in this question paper.
5. When this Assessment starts, **please immediately write your Student Number** below (using pen), and **shade your Student Number in the OCR form**. No extra time given at the end of the exam to do so!
6. This is a **Closed Book Assessment**.

STUDENT NUMBER:

A								

(Write your Student Number above legibly with a pen.)

Questions	Possible	Marks
Q1-10	**30**	
Q11	**15**	
Q12	**15**	
Q13	**20**	
Q14	**20**	
Total	**100**	

11. **Analysis Question [15 marks ~ 3 marks each]**

Prove (the statement is correct) or disprove (the statement is wrong) the following statements below. If you want to prove it, provide the proof or at least a convincing argument. If you want to disprove it, provide at least one counter example. 3 marks per each statement below (1 mark for circling true/false, 2 marks for explanation):

i. Given a BST (not a balanced BST) of N distinct integers, we delete an integer X that is guaranteed to exist in that BST. Then, we re-insert X again. The resulting structure of that BST will be exactly the same.

True / False

ii. Given an undirected and connected graph, if the weight of the shortest path from vertex s to vertex x, i.e $s\sim> x$, is smaller than the weight of the shortest path from s to every other vertex, and the weight of the shortest path from vertex t to vertex x, i.e $t \sim> x$ is smaller than the weight of the shortest path from t to every other vertex, then the shortest path from s to t must be $s \sim> x \sim> t$ in **all cases**.

True / False

iii. BST property (for a BST containing only unique integers) can be re-worded as follows: For every vertex x that is not a leaf, x.left.key < x.key if x has a left child and x.key < x.right.key if x has a right child.

True / False

iv. Given an undirected weighted graph with non-negative edges in adjacency matrix form, we cannot perform Dijkstra's faster than $O(V^2 * log\ V)$ time.

True / False

v. Given a min heap with 3 or more nodes, when performing an extractMin() method, the resulting shiftDown() call will always result in at least 1 swap.

True / False

12. **Hashing [15 marks]**

a) **[7 marks]** Consider inserting the keys 23, 9, 31, 18, 39, 29, 33 in sequence into a hash table of length *m = 13* with the hash function

$$h(key) = key \ \textbf{mod} \ m$$

Then we delete keys 31 and 33
If we use double hashing with the second hash function

$$h_2(key) = 1 + key \ \textbf{mod} \ 5$$

to solve the collision problem, show the contents of the hash table after performing all the above operations. How many probes are required to find out that the key value 81 is not in the resulting hash table?

Number of probes required = ___________

0	1	2	3	4	5	6	7	8	9	10	11	12

b) **[8 marks]** We have a hash table with size 11 using division method and quadratic probing to resolve collision. Show content of the table after inserting the following key values: 25, 18, 26, 19, 47, 36.

0	1	2	3	4	5	6	7	8	9	10

How many probes do you need if you try to search for 14. Explain your answer.

13. **A new ADT [20 marks]**

Given an array **A** of N ($1{,}000{,}000 \leq N \leq 1{,}000{,}000{,}000$) unsorted unique integers as input, we want an ADT that does the following operations in the stated time complexity. Read ALL required operations and think carefully before starting on the problem!

i. **initADT(A)** will initialize the ADT with the integers in array **A**. Run time is $\boldsymbol{O(N)}$.

ii. **replaceValue(h,j)** will replace the integer value h with integer value j. If h is not found in the ADT, then nothing happens. Run time is $\boldsymbol{O(logN)}$. *Assume that values in the ADT will always be unique regardless of any calls to **replaceValue**.*

iii. **replaceBlock(i,K)** will replace the integers from index i to index $i + (K.length - 1)$ of A with the integers in array **K**, where $\frac{N}{10} <= K.length <= \frac{N}{2}$. If $i + (K.length - 1) > N - 1$, nothing happens. Run time is $\boldsymbol{O(N)}$. *Assume that values in the ADT will always be unique regardless of any calls to **replaceBlock.***

iv. **findMin()** will find the min value in A and return it. Run time is $\boldsymbol{O(1)}$.

a) **[5 marks]** Give the algorithm for **initADT(A)** in the required time complexity. Detail clearly the data structure(s) you use in your implementation of the ADT, and what is/are the data to be stored in the data structure(s).

e.g data structure: sorted array
data stored: pair info (i,j) where j = A[i] for all i from 0 to N-1. 2nd field used for sorting.
// Algorithm for initADT here .....

b) **[5 marks]** Give the algorithm for **replaceValue(h,j)** in the required time complexity.

c) **[5 marks]** Give the algorithm for **replaceBlock(i,K)** in the required time complexity.

d) **[5 marks]** Give the algorithm for **findMin()** in the required time complexity.

14. **Z city transportation [20 marks]** (Read the question carefully!)

In a futuristic city Z, the transportation system is provided by 2 transportation companies, *Red* and *Blue*. The technology for transportation relies on a transportation tube which houses a high speed anti-gravity train which will take commuters in one direction from place A to place B. In order to get from place B back to place A, they will have to use another transportation tube that goes in the direction from B to A. **Assume for any 2 places A and B that has a tube going from A to B, there will be another tube going from B to A.**

Half of the transportation tubes in Z colored in blue are provided by *Blue* and the other half colored in red by *Red*, with no pattern as to which part of the transportation network are the blue tubes and which parts are the red tubes. Sometimes a blue tube by *Blue* connects place A and B in the direction A->B, while a red tube by *Red* can connect A and B in the direction B->A. The cost of using red and blue tubes will differ as they are set by their respective companies, and not all red tubes or all blue tubes will cost the same.

Assume you have the pricing for using every transportation tube in Z, which locations they connect, the direction they run and also their color (i.e which company they belong to) answer the following questions. Assume there are V $(10{,}000 \leq V \leq 100{,}000)$ locations in the city, and E number of tubes where $2 * (V - 1) \leq E \leq V^2 - V$. Assume you can always get from any place u to any place v in Z using the transportation system.

a) **[5 marks]** As a commuter and programmer, you want to write a method

> $\boldsymbol{lowestCost(u, v)}$ – which will return the lowest cost to go from place u to place v and back to u again.

Show how you would model the above mentioned transportation system as a graph (what is the graph type, what are the vertices, what are the edges, edge weights and other relevant information) and give the best algorithm you can think of for $\boldsymbol{lowestCost(u, v)}$.

b) **[8 marks]** As competition heats up between *Red* and *Blue*, they decided to offer a promotional deal which runs for a year. In this promotion, every day they will set one price among {$1, $2, $3, $4, $5} for the use of all their respective tubes for the day (*Red* and *Blue* will most probably set a different price each). This is good news for all commuters, but a commuter will only get the deal if they take ONLY the tubes of one company, either *Red* or *Blue*, for that day. You now write another method

> $\boldsymbol{bestTubeType(u, v)}$ – which will return the lowest cost tube type (red or blue) to use to get from place u to v and back to u. Assume there will be no tie. If there is no way to get from u to v and back to u using only 1 single tube type simply return "not possible".

Show how you will re-model the graph of the transportation system if necessary and then give the best algorithm you can think of for $\boldsymbol{bestTubeType(u, v)}$.

c) **[7 marks]** After the year is over, Red and Blue found their promotional deal to be a financial disaster. Thus they decide to go back to their original pricing plan. However, they will now on an hourly basis lower the price for the least popular tube (there will only be one for each company) of the previous hour in hopes of luring commuters to use the tube. This time you want to write the method

> $\boldsymbol{lowestCost2(u, v, p, A, B)}$ – which returns the lowest cost to get from place u to v and back to u, given the lowered price p of the least popular tube A->B (i.e the least popular tube connecting A to B in that direction) for the hour.

Give an algorithm for $\boldsymbol{lowestCost2(u, v, p, A, B)}$ which will run in O(1) time. You are allowed to perform a pre-processing step first (i.e before any calls to $\boldsymbol{lowestCost2}$), and if you do so give the best algorithm you can think of for the pre-processing step.

~~~ END OF PAPER ~~~