NATIONAL UNIVERSITY OF SINGAPORE

CS2040S—Data Structures and Algorithms

2020/2021 Semester 2

Time Allowed: 2 hours

INSTRUCTIONS TO STUDENTS

1. Please write your Student Number only. Shade the circles corresponding to your student number correctly. Do not write your name.
2. The assessment paper contains **ELEVEN (11) questions** and comprises **EIGHTEEN (18) pages** including this cover page.
3. Weightage of each question is given in square brackets. The maximum attainable score is 100.
4. This is a **CLOSED** book assessment, but you are allowed to bring **ONE** double-sided A4 sheet of notes for this assessment. You may not bring any magnification equipment! You may NOT use a calculator, your mobile phone, or any other electronic device.
5. Write all your answers in the space provided in the **ANSWER BOOKLET**.
6. You are allowed to write with pencils, as long as it is legible.
7. Shade the answer circles fully and make sure the shading is dark enough.
8. Read through the problems before starting. Do not spend too much time on any one problem.
9. For the multiple choice questions, no partial credit will be given. There is intended to be only one correct answer per multiple choice question.
10. For the open-ended questions, partial credit may be given, so show your work and explain your assumptions. You will be graded not only on the correctness of your answer, but also on the clarity with which you express it.

This page is intentionally left blank.

It may be used as scratch paper.

Question 1: True or False [9 marks]

For each of the following questions in this section, specify whether the statement is true or false.

A. Given a sorted array of n (comparable) elements, you can construct a balanced binary AVL tree in $\Theta(n)$ time in the worst-case. [1 mark]

B. One reason Bellman-Ford is better than Dijkstra is that it can find shortest paths in graphs with negative weight cycles, while Dijkstra's cannot. [1 mark]

C. A key advantage of a hash table over a Bloom Filter is that a hash table has no false negatives, while a Bloom Filter may suffer false negatives. (A "false negative" occurs when a search operation returns false, even though the "correct" answer is true.) [1 mark]

D. You can find the minimum item in a regular (unaugmented) AVL tree containing n elements in $\Theta(\log n)$ time, worst-case. [1 mark]

E. You can find the median item in a regular (unaugmented) AVL tree containing n elements in $\Theta(\log n)$ time, worst-case. [1 mark]

F. An adjacency list is more space efficient than an adjacency matrix for a graph where every node has (exactly) degree $d = \sqrt{n}$. [1 mark]

G. Considering an AVL tree as a graph, an AVL tree with n nodes has diameter $\Theta(\log n)$. [1 mark]

H. You are given a connected, undirected, unweighted graph G that contains $n > 3$ nodes. Node u is a node in the graph. The maximum shortest-path distance from u to any node in the graph is d. Thus the diameter of the graph is d. [1 mark]

I. Given a graph G and a node s in G, the shortest path tree rooted at s always includes the lightest edge in the graph G. (The shortest path tree rooted at s contains the shortest paths from s to every node in the graph.) [1 mark]

Question 2: Miscellaneous Questions [14 marks]

A. Assume you have an undirected graph G with 12 nodes and 6 edges. There are no cycles in the graph. How many connected components are there in the graph? [2 marks]

1. 9 connected components.
2. 8 connected components.
3. 7 connected components.
4. 6 connected components.
5. 5 connected components.
6. 4 connected components.
7. You cannot determine the exact number of connected components from the information given.

B. Assume you are given a special priority queue `PSpecial` that implements the following operations with the following costs, assuming there are n elements in the priority queue:

- `insert`: $\Theta(\sqrt{n})$
- `extractMin`: $\Theta(\sqrt{n})$
- `decreaseKey`: $\Theta(\log n)$
- `isEmpty`: $\Theta(1)$

What is the running time for Dijkstra's Algorithm on a connected graph with n nodes and m edges if it uses the `PSpecial` priority queue? [3 marks]

1. $\Theta(n\sqrt{n})$
2. $\Theta(m\log n)$
3. $\Theta(m\sqrt{n})$
4. $\Theta(m\sqrt{n}+n\log n)$
5. $\Theta(n\sqrt{m}+m\log n)$
6. $\Theta(n\sqrt{n}+m\log n)$

C. Choose the tightest possible bound for the following function:

$$T(n) = 24n^{0.5}\log^2(n) + 1.7\log^3 n$$

(Choose the smallest function that is an asymptotic upper bound.) [3 marks]

1. $O(1)$
2. $O(\log n)$
3. $O(n)$
4. $O(n\log n)$
5. $O(n\log^2 n)$
6. $O(n^2)$

D. What is the worst case running time for a `Find` operation on the Union-Find data structure that is using both the weighted union heuristic and the path compression heuristic? Assume there are at most n disjoint sets, initially each element is in its own set, and $\alpha(n)$ is the (single parameter version of the) inverse Ackermann function. [3 marks]

1. $\Theta(1)$
2. $\Theta(\alpha(n))$
3. $\Theta(\log n)$
4. $\Theta(\sqrt{n})$
5. $\Theta(n)$
6. None of the above.

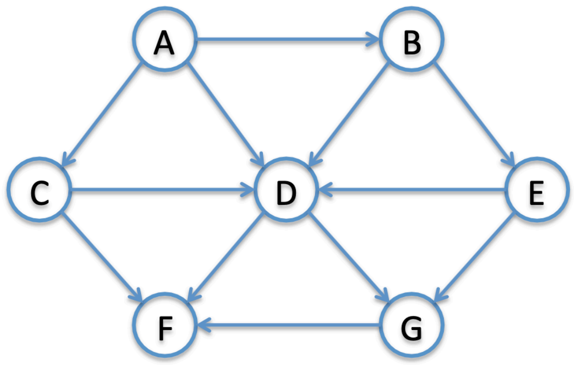
E. You are building a new university campus. You have already designed the buildings and laid out paths connecting the buildings. Currently, the paths are all uncovered, exposed to the rain.

Building a covered walkway is more expensive, and the cost of building a covered walkway depends on how long the path is. (A longer path costs more!) You want to make sure that there is a covered path from the main administrative building (otherwise known as "Building S") to every other building (so that the university president can visit everyone without getting wet in the rain). Which algorithm that we have studied can be used to determine the cheapest set of paths to cover? (Choose the most efficient option that satisfies this goal.) [3 marks]

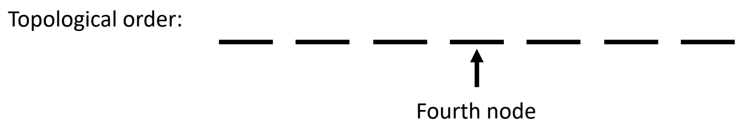
1. Breadth-First Search (BFS)
2. Depth-First Search (DFS)
3. Bellman-Ford
4. Dijkstra's Algorithm
5. Prim's Algorithm
6. DAG-shortest-paths
7. SteinerMST

Question 3: A Graph Question [3 marks]

Below is a directed acyclic graph (DAG) consisting of seven nodes. In this problem, you will run a topological sort on this DAG. Perform the topological sort by executing a post-order depth-first search on the graph. (Do not use an alternate technique for finding a topological order.)



Start your topological sort at node A. Assume that at every node, the adjacency list is sorted alphabetically from smaller to larger letters. (Thus, when enumerating edges adjacent to a node, always enumerate edges in alphabetical order, choosing smaller letters over larger letters.)



Topological order: __ __ __ __ __ __ __

Fourth node

What is the fourth node in the topological order? [3 marks]

1. A
2. B
3. C
4. D
5. E
6. F
7. G

Question 4: Hashing & Hash Tables [14 marks]

This is a problem about hash tables. Assume our hash table is 1-indexed, i.e., the first bucket is 1. There are 13 buckets in our hash table, and the last bucket is number 13. Consider the following hash function that maps elements to a table of size 13:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
2	13	8	9	7	1	2	4	1	11	5	7	2	8	5	2	9	5	3	10	5

Consider the following sequence of items added to the hash table:

F	R	O	G	S	Q	U	A	C	K

A. If the hash table resolves collisions via chaining, then how many elements are there in the longest linked list? [3 marks]

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. None of the above

B. If the hash table resolves collisions via linear probing (i.e., a form of open addressing), then in what slot is the 'A' placed? (Use 1-based indexing where the first slot of the array is numbered 1.) [3 marks]

1. 2
2. 3
3. 4
4. 5
5. 6
6. 7
7. 8
8. None of the above

C. Assume we have some hash table using open addressing with some hash function that satisfies the Uniform Hashing Assumption. There are n items that have been inserted in the table (and no items have been deleted). The table has m buckets. (The table does not grow or shrink.) Which of the following statements is true?

I. As long as $n \leq m$, the expected cost to search for an item is $O(1)$.

II. As long as $n \leq 2m/3$, the expected cost to search for an item is $O(1)$.

III. As long as $n \leq m/2$, the expected cost to search for an item is $O(1)$.

[2 marks]

1. None of the statements.
2. Statement III.
3. Statements II. and III.
4. Statements I. and II. and III.

D. You are designing a hash table with chaining. You are using a fixed sized array of size m, and each slot in the array contains a pointer to an AVL tree. (That is, instead of using a linked list for each slot, we are using an AVL tree.) When an item maps to slot j, we insert it into the AVL tree associated with slot j.

Assume that we have already inserted n items into the table where $n = 4m$. And assume that the hash function satisfies the simple uniform hashing assumption (SUHA). What is the expected time to search for an item in this table? [3 marks]

1. $\Theta(1)$
2. $\Theta(\log\log(n))$
3. $\Theta(\log(n))$
4. $\Theta(n)$
5. $\Theta(n \log n)$

E. Assume that resizing an array from size k_1 to size k_2 takes time $\Theta(\max(k_1, k_2))$. You are designing a table that supports inserts and deletes. Your table uses the following rule for resizing:

$$\begin{array}{lll} \text{if} & (n == m) & \text{then resize: } m \leftarrow 2m \\ \text{if} & (n <= \sqrt{m}) & \text{then resize: } m \leftarrow 2\sqrt{m} \end{array}$$

Assume the table starts out empty at size 100 and has a minimum size of 100. (It never shrinks below 100.) If you perform n operations on this table, then what is the amortized time for each operation? (Choose the smallest bound that is an asymptotic upper bound on the amortized time.) [3 marks]

1. $O(1)$
2. $O(\log\log n)$
3. $O(\log n)$
4. $O(\sqrt{n})$
5. $O(n)$
6. $O(n \log n)$

Question 5: Heaps of Fun [6 marks]

This question involves a new mergeable heap. To implement the mergeable heap, we begin with a regular binary heap, implemented using an array. The new MERGE operation is implemented as follows:

1. Initially, the two heaps are stored in arrays A_1 and A_2.
2. Concatenate the two arrays A_1 and A_2 by copying them both sequentially to a new array A_3. Copy the smaller array first and then the larger array. (E.g., if A_2 has fewer elements than A_1, then array A_3 contains the elements of A_2 before the elements of A_1.)
3. Build a new heap from the resulting array A_3.

The new mergeable heap supports four operations:

- `createHeap` creates an empty heap (in $O(1)$ time).
- `insert` adds a new item to an existing heap (using the regular binary heap insert).
- `extractMin` returns the minimum item from an existing heap (using the regular binary heap extractMin).
- `merge` creates a new heap by combining two existing heaps using the MERGE algorithm above.

A. If you have two of these new mergeable heaps of size n, what is the running time for this algorithm to merge the two heaps? [3 marks]

1. $\Theta(\log n)$
2. $\Theta(n)$
3. $\Theta(n \log n)$
4. $\Theta(n^2)$
5. None of the above.

B. What is the worst-case total runnning time for a sequence of n operations on the new mergeable heap? [3 marks]

1. $\Theta(\log n)$
2. $\Theta(n)$
3. $\Theta(n \log n)$
4. $\Theta(n \log^2 n)$
5. $\Theta(n^2)$
6. None of the above.

Question 6: Invariants [9 marks]

A. You are running Bellman-Ford on a directed, weighted graph where every node is reachable from a designated source s. Which of the following invariants is true for Bellman-Ford on this type of graph? (An estimate is correct if it is equal to the shortest distance to the source.)

I. After iteration k of the outer loop, every node within k hops of the source has a correct estimate.

II. After iteration k of the outer loop, every node whose shortest path from the source is $\leq k$ hops has a correct estimate.

III. After iteration k of the outer loop, every node whose estimate has been decreased k times has a correct estimate.

[3 marks]

1. Statement I.
2. Statement II.
3. Statement III.
4. Statements I and II.
5. Statements I and III.
6. Statements II and III.
7. Statements I, II and III.
8. None of the statements.

B. You are running Dijkstra's Algorithm on a directed, weighted graph with no negative weights and source node s. Every node is reachable from the source s.

I. Assume u has been extracted from the priority queue at least once. Then the estimate at u is equal to the distance from s to u.

II. At all times, the estimate at u is $\leq$ the distance from s to u.

III. At all times, the estimate at u is $\geq$ the distance from s to u.

[3 marks]

1. Statement I.
2. Statement II.
3. Statement III.
4. Statements I and II.
5. Statements I and III.
6. Statements II and III.
7. Statements I, II and III.
8. None of the statements.

C. You are running Dijkstra's Algorithm on a directed, weighted graph with no negative weights and source node s. Every node is reacahble from the source s.

I. Throughout the execution, you relax each edge exactly once.

II. Through the execution, you update the estimate at a node v at most once.

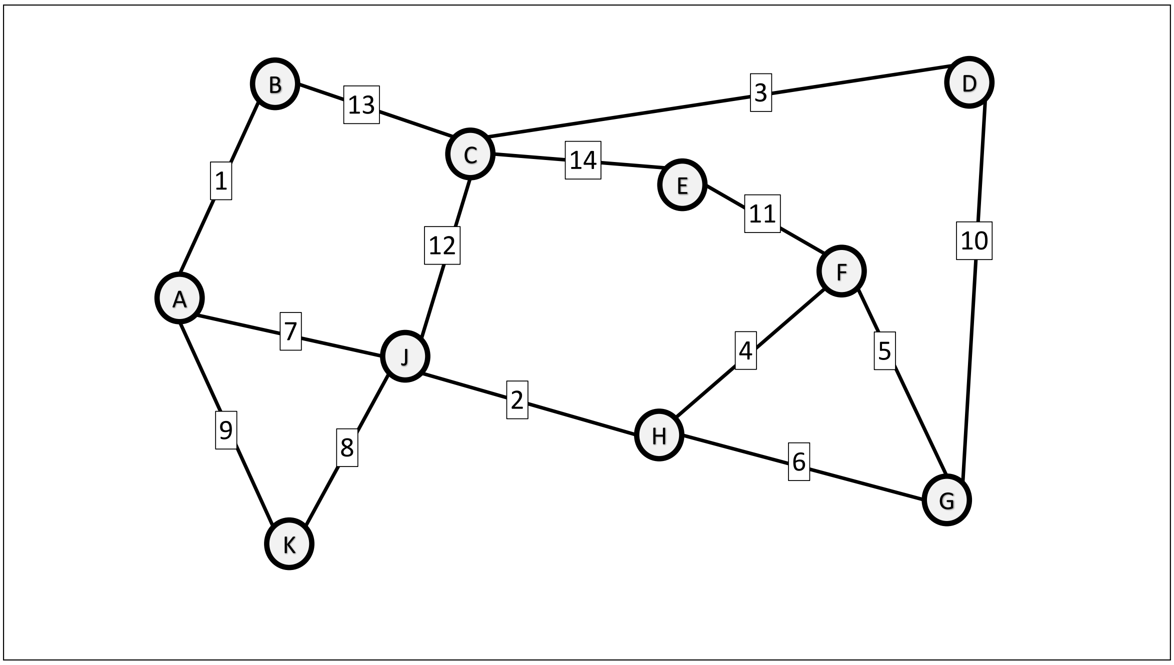
III. Assume that at some point t in the execution, you extract a node u from the priority queue with estimate d. If you extract node v from the priority queue at some point after t, then it has an estimate of at least d.

[3 marks]

1. Statement I.
2. Statement II.
3. Statement III.
4. Statements I and II.
5. Statements I and III.
6. Statements II and III.
7. Statements I, II and III.
8. None of the statements.

Question 7: Minimum Spanning Trees [13 marks]

All the questions in this section refer to this connected, undirected, weighted graph with 10 nodes and 14 edges (each with a distinct weight from the interval $[1, 14]$).



Assume that we have found a minimum spanning tree (MST) for the undirected, weighted, connected graph in the figure above. For each of the statements below, indicate whether it is true or false. (Note that a statement is true only if it correctly identifies whether or not the edge is in the MST and if the because part is a sufficient reason for why the edge is or is not in the MST.)

A. True or false: edge (A, J) is in the MST because it is the lightest edge on the cycle (A, J, K).
[1 mark]

B. True or false: edge (H, J) is in the MST because it is the lightest edge adjacent to node J.
[1 mark]

C. True or false: edge (E,F) is not in the MST because it is the heaviest edge across the cut separating node F from the rest of the graph. [1 mark]

D. True or false: edge (E,F) is in the MST because there is some cut for which (E,F) is the lightest edge across the cut. [1 mark]

E. True or false: edge (F,H) is in the MST because it is the lightest edge on the cycle (F,G,H). [1 mark]

F. True or false: edge (C,E) is not in the MST because it is the heaviest edge in the graph. [1 mark]

G. True or false: edge (C,E) is not in the MST because there exists a path from C to E where every edge is lighter than the weight of (C,E). [1 mark]

H. Assume we find an MST of this graph by running Kruskal's Algorithm. Which is the first edge considered by the algorithm that is not part of the eventual MST? [2 marks]

1. (G,H)
2. (A,J)
3. (J,K)
4. (A,K)
5. (D,G)
6. (E,F)
7. (C,J)
8. (B,C)

I. Assume we find an MST of this graph by running Prim's Algorithm starting with node C. Which is the second edge added to the MST? [2 marks]

1. (D,G)
2. (A,B)
3. (J,H)
4. (B,C)
5. (C,E)
6. (C,J)
7. (C,D)

J. Assume we find an MST of this graph by running Boruvka's Algorithm. After one Boruvka Step, how many connected components are there in the graph? [2 marks]

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8

Question 8: Shortest Paths [11 marks]

A. You are running Bellman-Ford on a connected, weighted, directed graph G where all the edges have weight 0 or 2. You modify the algorithm to relax all the edges of weight 0 before any of the edges of weight 2 (within each iteration). True or false: It is always the case that all the estimates are correct after 2 iterations of the outer loop. [2 marks]

B. You are running Dijkstra's Algorithm on a weighted, directed graph where all the edges have weight 0 or 2. There is a specially designated source s and all the nodes are reachable from the source. We use a special priority queue which has the following property:

> If there are k distinct priority values in the priority queue, then the cost of every operation is $O(\log k)$

(For example, if every item in the priority queue has priority 7, then all operations have cost $O(1)$.) We modify Dijkstra's Algorithm so that we do not store nodes in the priority queue that have infinite estimates: we only store them in the priority queue once they have a finite estimate. (Otherwise, Dijkstra's Algorithm works as usual, extracting the node with the minimum estimate from the priority queue and relaxing its outgoing edges.) What is the worst-case running time for Dijkstra's Algorithm on such a graph with n nodes and m edges if it uses this priority queue? [3 marks]

1. $\Theta(n)$
2. $\Theta(m)$
3. $\Theta(m \log n)$
4. $\Theta(n \log m)$
5. $\Theta(nm)$
6. $\Theta(n^3)$

C. You are running Prim's Algorithm on a connected, weighted graph where all the edges have weight 0 or 2. We use a special priority queue which has the following property:

> If there are k distinct priority values in the priority queue, then the cost of every operation is $O(\log k)$

(For example, if every item in the priority queue has priority 7, then all operations have cost $O(1)$.) We modify Prim's Algorithm to not store nodes in the priority queue with infinite priority. What is the running time for Prim's Algorithm on such a graph with n nodes and m edges if it uses this priority queue? [3 marks]

1. $\Theta(n)$
2. $\Theta(m)$
3. $\Theta(m \log n)$
4. $\Theta(n \log m)$
5. $\Theta(nm)$
6. $\Theta(n^3)$

```
modifiedBFS(nodes V, edges E, weights w, source s)
    let Q be a queue
    for all nodes v in V: visited[v] = false
    for all nodes v in V: est[v] = emptyset
    est[s] = {0}
    Q.add(s)

    while !Q.empty():
        u = Q.dequeue()
        if (!visited[u]):
            visited[u] = true
            for all edges (u,v) in E:
                for all d in est[u]:
                    if (d + w(u,v) not in est[v]):
                        add (d + w(u,v)) to est[v]
                        visited[v] = false
                if !visited[v] then Q.enqueue(v)
```

D. Bob McBobface decides to design a new algorithm for finding all the possible distances between two nodes in a directed acyclic graph. He develops a modified version of Breadth-First-Search that integrates ideas from Bellman-Ford: each node stores a set of estimates (instead of just one), and whenever the BFS visits a node, it relaxes its outgoing edges. If a node u has a new estimate added to its set, then it is marked unvisited, and added to the BFS queue. The pseudocode for his algorithm is above.

What is the worst-case running time of this algorithm on a weighted, directed acyclic graph with n nodes and m edges? Every node is reachable from the source s. (You may assume that the algorithm finds the correct answer in the following sense: if it terminates, then the set of estimates at every node contains all possible distances from the source to that node. There is no restriction on the size of the weights.) Choose the smallest asymptotic running time that is an upper bound on the running time, or none of the above if none of the answers is a bound on the worst-case running time. [3 marks]

1. $O(m)$
2. $O(m \log n)$
3. $O(mn)$
4. $O(m^2 n)$
5. $O(2^m)$
6. None of the above.

Question 9: JohnnySort [5 marks]

Consider the following sorting algorithm invented by our friend Johnny:

```
void JohnnySort(int[] A){
  int size = A.length;
   for (int i=size-1; i>=0; i--) {  // Outer loop
     for (int j=i; j<size-1; j++) { // Inner loop
       if (A[j] >= A[j+1]) {
         int temp = A[j+1];
         A[j+1] = A[j];
         A[j] = temp;
       }  // End of if statement
     }  // End of inner loop
   }  // End of outer loop
}
```

Please give an invariant that is sufficient to show that the algorithm sorts the array correctly. Your invariant should be of the form: “After each iteration of the outer loop: ...”

[5 marks]

Question 10: Algorithm Design [16 marks]

[16 marks]

In this section, please give short and clear answers to each of the questions. If your answers are not clear, you will not get credit—even if your underlying idea is correct! Part of the exam is testing whether you can explain yourself clearly. (We will not accept further explanations after the exam is over.) For example, if we cannot read your handwriting, we will not understand your idea. If you write too much, we may not be able to figure out which part is your answer. If you give several different answers or explanations, we may not be able to determine which one you intend. Please give answers that are short and clear.

The questions in this section do not have individual marks specified for each subpart. The marks given will depend on the algorithm, its efficiency, and how well it is explained.

After 2020, a lot of people want to get back to travelling once it is safe! Singapore Air has designed a round-the-world plan: you pay a fixed price for your round-the-world ticket, and you can take as many flights as you want, subject to four constraints:

1. Your first flight has to leave from Singapore;
2. Your last flight has to end in Singapore;
3. You can only leave Singapore once; and
4. Every flight must land in a destination to the east of where it departed from. (No going west!)

Assume you are given a map with n cities and m flights (each going from west to east). You are also given the length of each flight. (You may assume that you are given this information in a reasonably efficient format.) Since you hate wasting time on planes, you want to find a round-the-world trip that minimizes the total time on flights. Design an algorithm for efficiently solving this problem. (Your solution should be as fast as possible.)

A. Give a brief two sentence explanation of your solution. To help explain your algorithm, please draw a graph containing 5 nodes and illustrate how your algorithm works using that graph. (If your algorithm is simple, the resulting illustration should also be simple. If your algorithm is more complicated, you may need to draw more.)

B. If the map has n cities and m flights (each going from west to east), what is the running time of your algorithm? (Give a worst-case asymptotic bound as a function of n and m that is as tight as possible.)

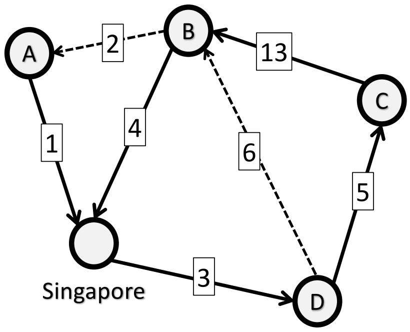
C. Briefly (in one sentence) explain the running time you stated in Part 10B.

D. The airline decides to modify its policy: you may take flights that go in the wrong direction, from east to west. However, you will be charged an additional SGD 500 for each flight that goes from east to west. Thus you want to minimize the number of east-to-west (wrong way) flights that you take.

You decide that you do not mind a long trip, but you do have a deadline: you have to get back to Singapore in time for the next semester! Given a time bound T, your goal is to find a route from Singapore back to Singapore that takes time $\leq T$ and minimizes the monetary cost, i.e., minimizes the number of flights in the wrong direction from east to west. (The goal is not to minimize the time, only to ensure that the total flight time is $\leq T$. The goal is to minimize the number of flights that go west.)

You are given a time T, and you are given a map with n cities and m flights, where each flight is labelled with a length and also whether it is the "right way" from west-to-east or the "wrong way" from east-to-west. (Please do not worry today about such issues as the geography or topology of the earth and whether a given set of flight times is reasonable or whether the flights go anywhere in particular.)

For example, you might have the following map containing Singapore and four other cities:



In this example, the edges (B,A) and (D,B) are "wrong-way" east-to-west edges. All the other edges are west-to-east edges. If you need to meet a deadline $T = 15$, then the route from Singapore to D to B to Singapore is ideal: it takes time 13 and crosses only one "wrong-way" edge. Any route that cross zero "wrong-way" edges costs more than 15, so this is the best solution.

Give a brief four sentence explanation of your algorithm. (It is sufficient for your algorithm to return the length of the route, which is $\leq T$, and the number of wrong-way flights.)

E. Illustrate how your algorithm would work using the example graph on the answer sheet. If your algorithm constructs any new graphs along the way, please draw those as well. In this example, the edges (B,A) and (D,B) are "wrong-way" east-to-west edges. All the other edges are west-to-east edges. (If you algorithm is simple, the resulting illustration should also be simple. If your algorithm is more complicated, you may need to draw more.)

F. Explain why your algorithm works, giving the minimum necessary to convince that your algorithm is correct.

G. If the map has n cities and m flights (some goine east to west and some going west to east), what is the running time of your algorithm? (Give a worst-case asymptotic bound as a function of n and m that is as tight as possible.)

H. Briefly explain the running time you stated in Part 10G.

Question 11: Just for fun: The Blue-Eyed Islanders [0 marks]

[0 mark]

There is an island upon which a tribe resides. The tribe consists of 1000 people, with various eye colours. Yet, their religion forbids them from knowing their own eye color, or even from discussing the topic. Thus, each resident can (and does) see the eye colors of all other residents, but has no way of discovering his or her own (there are no reflective surfaces).

If a tribesperson does discover his or her own eye color, then their religion compels them to say goodbye to every other islander and leave the island forever by noon the following day. All the tribespeople are highly logical and devout, and they all know that all the others are also highly logical and devout (and they all know that they all know that all the others are highly logical and devout, and so forth).

Of the 1000 islanders, it turns out that 100 of them have blue eyes and 900 of them have brown eyes, although the islanders are not initially aware of these statistics (each of them can of course only see 999 of the 1000 tribespeople).

One day, a blue-eyed foreigner visits the island and wins the complete trust of the tribe. One evening, he addresses the entire tribe to thank them for their hospitality. However, not knowing the customs, the foreigner makes the mistake of mentioning eye color in his address, remarking "how unusual it is to see another blue-eyed person like myself in this region of the world".

What effect, if anything, does this faux pas have on the tribe?

Courtesy of Terence Tao:
http://terrytao.wordpress.com/2011/04/07/the-blue-eyed-islanders-puzzle-repost/.

Nothing to submit for this question. This is for you to think about after the exam. You can post your thoughts and ideas in the class forum.

— END OF PAPER —

Scratch Paper

— HAPPY HOLIDAYS! —

CS2040S — Data Structures and Algorithms
School of Computing
National University of Singapore

Final Assessment — Answer Sheet

2020/2021 Semester 2

Time allowed: 2 hours

Instructions (please read carefully):

1. Write your **student number** on the right and, using ink or pencil, shade the corresponding circle **completely** in the grid for each digit or letter. DO NOT WRITE YOUR NAME!
2. This answer booklet comprises **ELEVEN (11) pages**, including this cover page.
3. All questions must be answered in the space provided; no extra sheets will be accepted as answers. You may use the extra page behind this cover page if you need more space for your answers.
4. You must submit only the **ANSWER SHEET** and no other documents. The question set may be used as scratch paper.
5. An excerpt of the question may be provided to aid you in answering in the correct box. It is not the exact question. You should still refer to the original question in the question booklet.
6. You are allowed to use pencils, ball-pens or fountain pens, as you like as long as it is legible (no red color, please).
7. **Marks may be deducted** for i) unrecognisable handwriting, and/or ii) excessively long explanations.
8. Each multiple choice question is intended to have only one answer. Please fill in the appropriate bubble.

STUDENT NUMBER

A									
U○	0	0	0	0	0	0	0	A	N
A●	1	1	1	1	1	1	1	B	R
HT○	2	2	2	2	2	2	2	E	U
NT○	3	3	3	3	3	3	3	H	W
	4	4	4	4	4	4	4	J	X
	5	5	5	5	5	5	5	L	Y
	6	6	6	6	6	6	6	M	
	7	7	7	7	7	7	7		
	8	8	8	8	8	8	8		
	9	9	9	9	9	9	9		

For Examiner's Use Only

Question	Marks
Q1	/ 9
Q2	/ 14
Q3	/ 3
Q4	/ 14
Q5	/ 6
Q6	/ 9
Q7	/ 13
Q8	/ 11
Q9	/ 5
Q10	/ 16
Q11	/ 0
Total	/100

This page is intentionally left blank.
Use it ONLY if you need extra space for your answers, and indicate the **question number clearly** as well as in the original answer box. **Do NOT** use it for your rough work.

Question 1A Construct balanced binary AVL tree in $\Theta(n)$ time. [1 marks]

○ True ○ False

Question 1B Bellman-Ford is better than Dijkstra. [1 marks]

○ True ○ False

Question 1C Hash table has no false negatives; Bloom Filter may have false negatives. [1 marks]

○ True ○ False

Question 1D Can find minimum in AVL tree in $\Theta(\log n)$ time. [1 marks]

○ True ○ False

Question 1E Can find median AVL tree in $\Theta(\log n)$ time. [1 marks]

○ True ○ False

Question 1F Adjacency list is more space efficient than an adjacency matrix. [1 marks]

○ True ○ False

Question 1G AVL tree with n nodes has diameter $\Theta(\log n)$. [1 marks]

○ True ○ False

Question 1H Diameter of the graph is d. [1 marks]

○ True ○ False

Question 1I Shortest path tree rooted always includes the lightest edge. [1 marks]

○ True ○ False

Question 2A How many connected components? [2 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

Question 2B Running time of Dijkstra's Algorithm using the PSpecial priority queue? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6

Question 2C Tightest bound? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6

Question 2D Worst case time for a Find operation? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6

Question 2E Which algorithm should we use? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

Question 3 What is the fourth node in the topological order? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

Question 4A Longest linked list? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

Question 4B What slot is ‘A’ in? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8

Question 4C Which is true? [2 marks]

○ 1 ○ 2 ○ 3 ○ 4

Question 4D What is the expected lookup time? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5

Question 4E What is the amortized time for each operation? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6

Question 5A Running time to merge two heaps? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5

Question 5B Worst-case total running time for *n* operations? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6

Question 6A Bellman-Ford Invariant? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8

Question 6B Dijkstra Invariant? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8

Question 6C Dijkstra Invariant? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8

Question 7A edge (A,J) is in the MST. [1 marks]

○ True ○ False

Question 7B edge (H,J) is in the MST. [1 marks]

○ True ○ False

Question 7C edge (E,F) is not in the MST. [1 marks]

○ True ○ False

Question 7D edge (E,F) is in the MST. [1 marks]

○ True ○ False

Question 7E edge (F,H) is in the MST. [1 marks]

○ True ○ False

Question 7F edge (C,E) is not in the MST. [1 marks]

○ True ○ False

Question 7G edge (C,E) is not in the MST. [1 marks]

○ True ○ False

Question 7H Which edge is the first found by Kruskal's that is not in the MST? [2 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8

Question 7I Which edge is the second added by Prim's to the MST? [2 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7

Question 7J How many connected components? [2 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8

Question 8A All the estimates are correct after 2 iterations. [2 marks]

○ True ○ False

Question 8B Worst-case running time of Dijkstra's Algorithm using this priority queue? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6

Question 8C Running time of Prim's Algorithm using this priority queue? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6

Question 8D Running time of modified BFS? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6

Question 9 [5 marks]

A good invariant for JohnnySort:

Question 10A

[?? marks]

A two sentence summary:

Picture:

Question 10B

[?? marks]

Asymptotic running time as a function of *n* and *m*:

Question 10C

[?? marks]

Running time explanation:

Question 10D

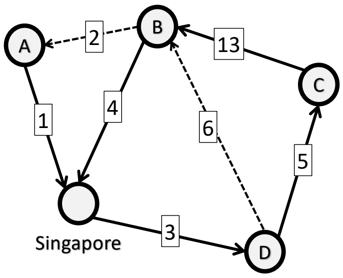
[?? marks]

A four sentence summary:

Question 10E

[?? marks]

Illustrate how your algorithm works using this example graph. In this example, the edges (B,A) and (D,B) are "wrong-way" east-to-west edges. All the other edges are west-to-east edges.



Question 10F

[?? marks]

Explain why your algorithm works:

Question 10G

[?? marks]

Asymptotic running time as a function of *n* and *m*:

Question 10H

[?? marks]

Running time explanation:

This page is intentionally left blank.
Use it ONLY if you need extra space for your answers, and indicate the **question number clearly** as well as in the original answer box. **Do NOT** use it for your rough work.

— END OF ANSWER SHEET —

Question 1A Construct balanced binary AVL tree in $\Theta(n)$ time. [1 marks]

○ True ○ False True

Question 1B Bellman-Ford is better than Dijkstra. [1 marks]

○ True ○ False False

Question 1C Hash table has no false negatives; Bloom Filter may have false negatives. [1 marks]

○ True ○ False False

Question 1D Can find minimum in AVL tree in $\Theta(\log n)$ time. [1 marks]

○ True ○ False True

Question 1E Can find median AVL tree in $\Theta(\log n)$ time. [1 marks]

○ True ○ False False

Question 1F Adjacency list is more space efficient than an adjacency matrix. [1 marks]

○ True ○ False True

Question 1G AVL tree with n nodes has diameter $\Theta(\log n)$. [1 marks]

○ True ○ False True

Question 1H Diameter of the graph is d. [1 marks]

○ True ○ False False

Question 1I Shortest path tree rooted always includes the lightest edge. [1 marks]

○ True ○ False False

Question 2A How many connected components? [2 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 4 : 6

Question 2B Running time of Dijkstra's Algorithm using the PSpecial priority queue? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 6 : $O(n)$

Question 2C Tightest bound? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 $3 : O(n)$

Question 2D Worst case time for a Find operation? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 $3 : O(\log n)$

Question 2E Which algorithm should we use? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 5: Prim's

Question 3 What is the fourth node in the topological order? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 5 : E.

Question 4A Longest linked list? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 4

Question 4B What slot is 'A' in? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 3: 4

Question 4C Which is true? [2 marks]

○ 1 ○ 2 ○ 3 ○ 4 3: II and III

Question 4D What is the expected lookup time? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 1: $\Theta(1)$

Question 4E What is the amortized time for each operation? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 1: $O(1)$

Question 5A Running time to merge two heaps? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 $3 : O(n)$

Question 5B Worst-case total running time for n operations? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 $5 : \Theta(n^2)$

Question 6A Bellman-Ford Invariant? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 2 : II or NONE (if you assume negative weight cycles)

Question 6B Dijkstra Invariant? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 4 : I and III

Question 6C Dijkstra Invariant? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 4 : I and III

Question 7A edge (A,J) is in the MST. [1 marks]

○ True ○ False False

Question 7B edge (H,J) is in the MST. [1 marks]

○ True ○ False True

Question 7C edge (E,F) is not in the MST. [1 marks]

○ True ○ False False

Question 7D edge (E,F) is in the MST. [1 marks]

○ True ○ False True

Question 7E edge (F,H) is in the MST. [1 marks]

○ True ○ False False

Question 7F edge (C,E) is not in the MST. [1 marks]

○ True ○ False False

Question 7G edge (C,E) is not in the MST. [1 marks]

○ True ○ False True

Question 7H Which edge is the first found by Kruskal's that is not in the MST? [2 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 1: (G,H)

Question 7I Which edge is the second added by Prim's to the MST? [2 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 1: (D,G)

Question 7J How many connected components? [2 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 3: 3

Question 8A All the estimates are correct after 2 iterations. [2 marks]

○ True ○ False False

Question 8B Worst-case running time of Dijkstra's Algorithm using this priority queue? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 $2 : \Theta(m)$

Question 8C Running time of Prim's Algorithm using this priority queue? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 $2 : \Theta(m)$

Question 8D Running time of modified BFS? [3 marks]

○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 $5 : O(2^m)$.

Question 9 [5 marks]

A good invariant for JohnnySort:

The subarray from $A[i]$ up to $A[size-1]$ contains the $size-i+1$ largest elements in the array.

Question 10A

[?? marks]

A two sentence summary:

The graph forms a DAG, so we can find the shortest paths using the algorithm for shortest paths on a DAG (i.e., toposort and relax the outgoing edges of nodes in topological order).

Picture:

Pretty picture of a directed acyclic graph.

Question 10B

[?? marks]

Asymptotic running time as a function of n and m:

$O(n+m)$ or $O(m)$.

Question 10C

[?? marks]

Running time explanation:

The running time is $O(n+m)$ (or $O(m)$) because the underlying graph is a directed acyclic graph with $|V| = n$ nodes and $|E| = m$ edges, and shortest path on such a DAG runs in $O(V+E)$ time (by performing a toplogical sort and relaxing the edges in order from beginning to end).

Question 10D

[?? marks]

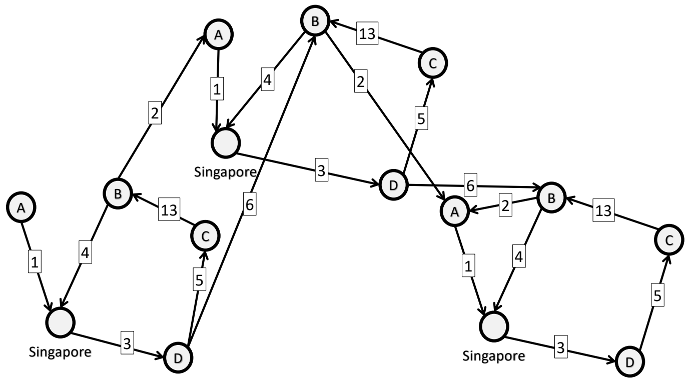
A four sentence summary:

If the graph contains W wrong-way flights, create W copies of the initial graph. Within a copy j, if edge (u,v) is a wrong-way edge, delete it from copy j and connect u in copy j with v in copy $j+1$. The result is a DAG, where a path that ends in copy j has crossed $j-1$ wrong-way edges. Check the distance estimate to Singapore in each of the copies, and return the estimate from the smallest copy that is $\leq T$.

Question 10E

[?? marks]

Illustrate how your algorithm works using this example graph. In this example, the edges (B,A) and (D,B) are "wrong-way" east-to-west edges. All the other edges are west-to-east edges.



Question 10F

[?? marks]

Explain why your algorithm works:

The algorithm works because the resulting graph is a directed acyclic graph, since we redirected all the “wrong-way” edges from one copy of the graph to the next. If you start in copy 1, the distance to Singapore in copy j specifies exactly the distance to Singapore if you are only willing to use $j-1$ “wrong-way” edges.

Question 10G

[?? marks]

Asymptotic running time as a function of n and m:

$O(m(n+m))$ or $O(m^2)$.

Question 10H

[?? marks]

Running time explanation:

The algorithm constructs a directed acyclic graph, and the new graph increases by at most a factor of m, as we have at most m copies of the original graph. Thus the new graph has $|V| \leq mn$ and $|E| \leq m^2$ and so the running time of the shortest path algorithm is $O(mn + m^2) \leq O(m^2)$. It then takes a further $O(m)$ time to check the solutions.